Computer Science & Engineering Department

Southern University Agriculture & Mechanical College

Software Engineering Project

Blood Bank Management System

Requirement & Specification

Design & Implementation Phase

Submitted by

Software Chasers Team

ADVISOR:

N. Gwee, PhD (Computer Science), PhD (Musicology)

Professor, Department of Computer Science

JUNE 2022

Table of Contents

Abstract

The Blood Bank Management System is an application that stores, processes, retrieves, and analyzes data about blood bank administration. It also supervises blood inventory management and other blood bank-related activities. The major goal of the blood bank management system is to keep track of blood, donors, blood groups, blood banks, and stock information. It keeps track of all information concerning blood, blood cells, stocks, and blood. Because the project is all done at the administrative level, only the administrator can see it.

Blood Bank Management System (BBMS) is a browser-based system that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way. Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassle free and corruption free and make the system of blood bank management effective.

It is developed in a manner that is easily manageable, time saving and relieving one from manual works. The requirement of the blood must be requested, and we supply the information of the donor. The donors can update their status whether they are available or not. This web-based application allows hospitals in Baton Rouge to make inventories of their blood bags online, subsequently, allowing particularly five hospitals to check the availability of blood bags anytime.

Requirement Phase

The list below is the blood bank management system at **Requirement Phase:**

Name of group:

The Name of the group we have decided,

Software Chasers

Name of software:

The Name of the Software we planned to do for this course is,

Blood Bank Management System

Target Audience:

Target Audience refer to the specific group of consumers most likely to want our product of service, in our case we target the,

Hospitals and Blood Bank Communities

Type of Software:

The Software we are going to use for our project is,

Web-based Application

Software & Hardware Requirements:

Software System requirements:

A Software requirements specification (SRS) is also known as System requirements Specification is a document or set of documentation that describes the features and behavior of a System or Software application.

Below the Software configurations must needed to run our project,

Software Configuration:

- Web Server **– Google Chrome - Version 102.0.5005.63 (Official Build)**

- Operating System **–** Microsoft **Windows 10 Home Edition**

Hardware System requirements:

The Hardware requirements are the requirement of a hardware device, w**ithout system requirements, the project will be a failure and the company will fail**. If the customer has system requirements that will satisfy their needs, then that company will have a higher percentage of success.

Below the Hardware configurations must needed to run our project,

Hardware Configuration:

- Processor: **Intel Dual core i3-5005U CPU**
- Hard Disk Drive **– 20 GB HDD**
- RAM **2GB**
- Monitor **– 15" digital color monitor**
- Display Type **– VGA**

Project Goal:

This website acts like a medium/an intermediary between hospitals and donors/receivers.

This project is created for two users.

- User1 – Hospitals/Clinics (Doctors)
- User2 – Donors/Receivers

Key Features:

These are the below the features we are going to include in our software,

- Register
- Home
- About

- Contact
- My Account
- Blood Info
- Stock Of Blood
- Blood Available
- Blood Requests
- Need Blood
- Status Of Request
- Blood Donation Request

Delimitations:

- We are not able to include the map of each location of Hospitals, for this we need an autentication token from Google maps and which is not free.
- We are not going to implement the Blood Info, Stock of Blood, Blood Requests, Need Blood, Status of Request, Blood Donation Request.

Specifications Phase

The list below is the blood bank management system at **Specification Phase:**

- Use Case Diagram
- Rapid Prototype
- Testing Protocol

Use Case Diagram:

The essential part of UML Diagrams of Blood Bank Management System Project. Each of these UML diagrams has role and function in developing Blood Bank System. The blood bank's UML Diagrams serves as the blueprint of the system development. It has the capability to let the programmers understand the work of a project.

This also enhance their knowledge and guides them on what to put into the project. In our project we have two users,

User1 – Hospitals/Clinics (Doctors)

After Hospital User login into his/her account, User can access to only below features,

Registration/Login/Logout

Home

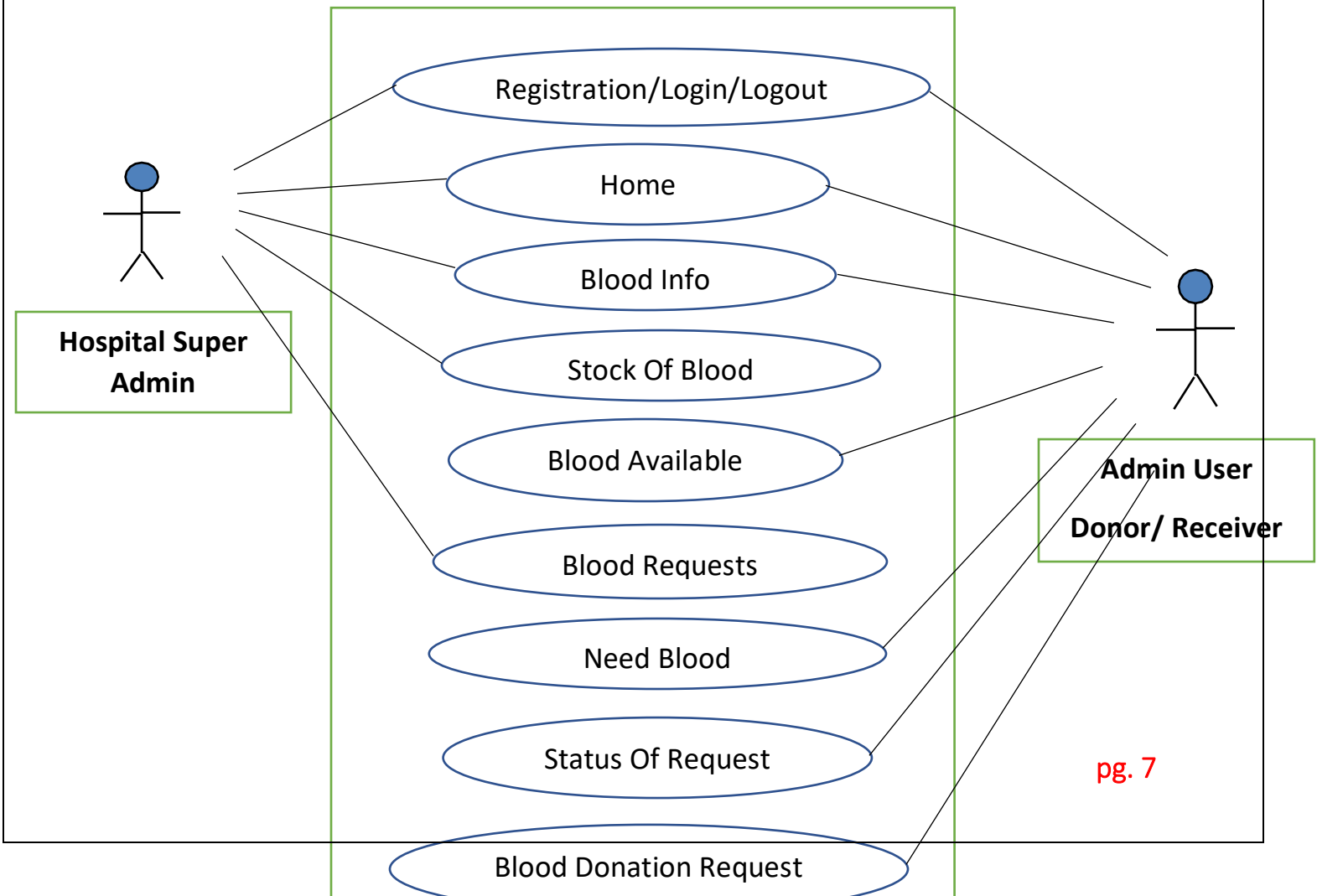
Blood Info

Stock Of Blood

User2 – Donors/Receivers

After Donor/Receiver login into his/her account, User can access to only below features,

Registration/Login/Logout, Blood Info, Stock of Blood, Blood Available, Blood Requests, Need Blood, Status of Request, Blood Donation Request.

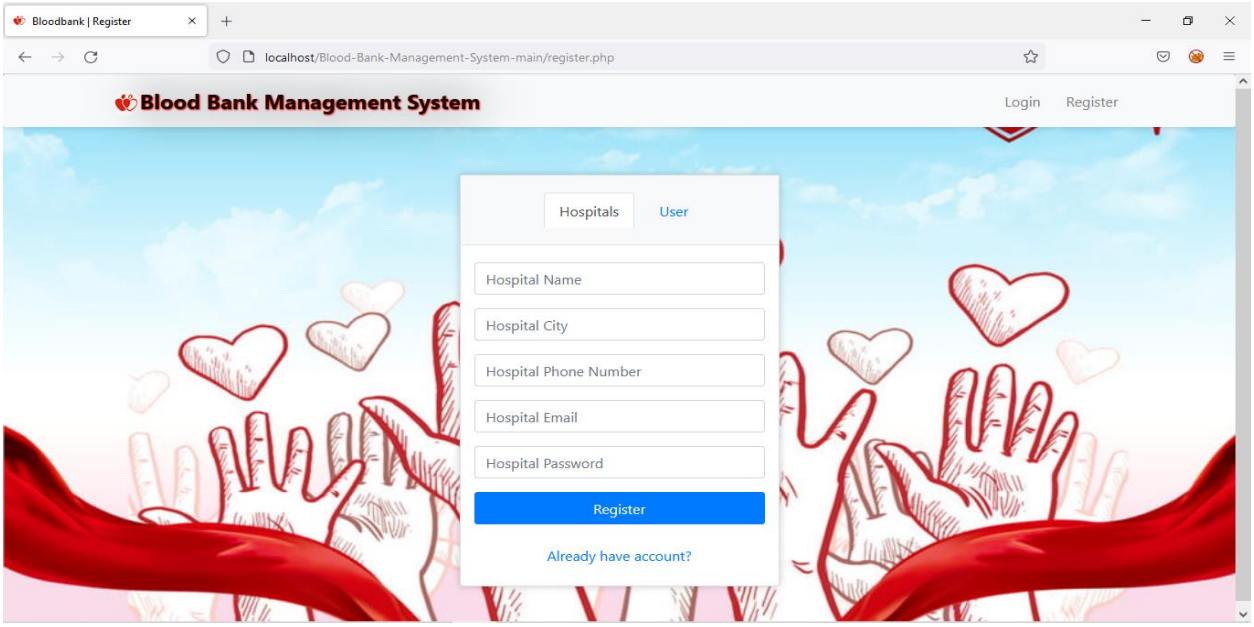


Rapid Prototype:

Rapid Prototype for Hospital Register Feature:

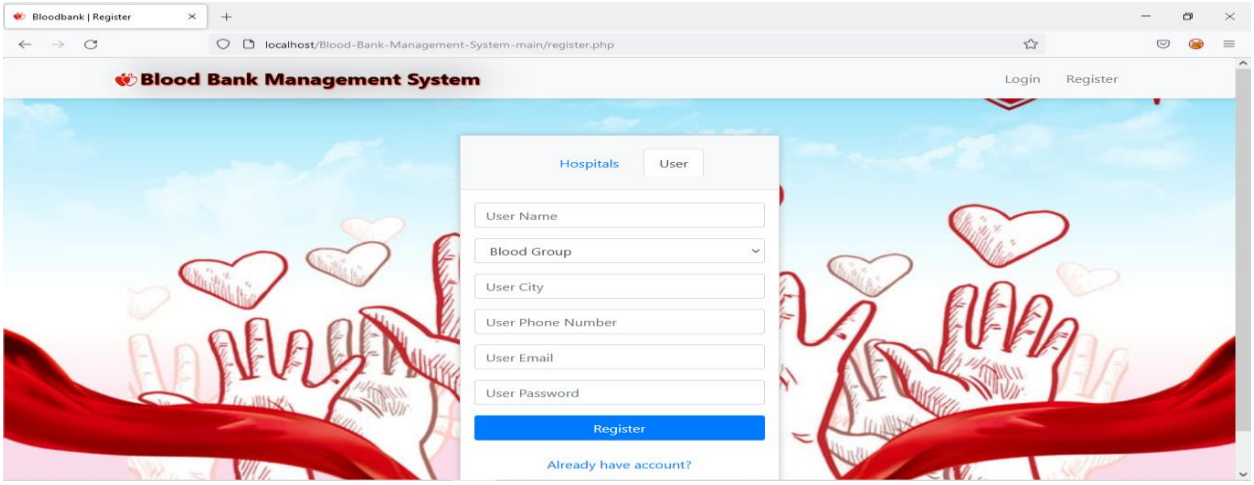
Fig1: Use case Diagram for Blood Bank Management System

This function allows the Hospital User to register as a new user with including the fields such as Hospital Name, Hospital City, Hospital Phone Number, Hospital Email and Hospital Password. Hospital user must remember the Username and Password to login into their account.



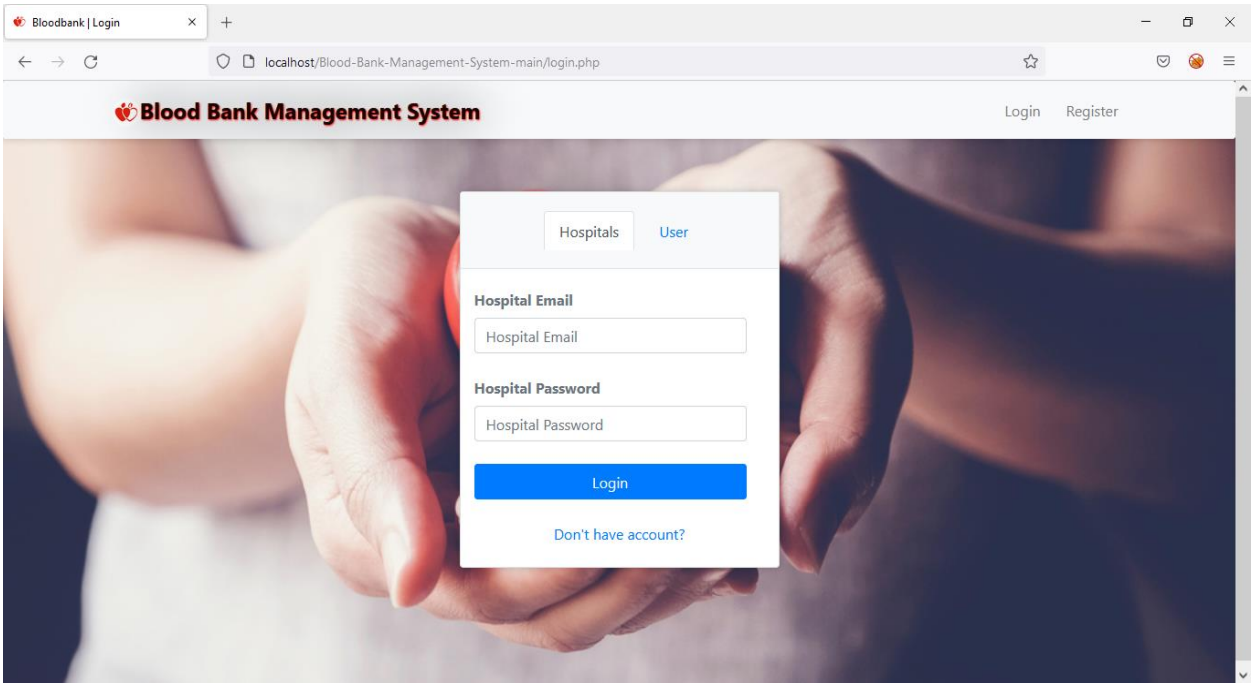
Rapid Prototype for User Register Feature:

This feature allows the User (Donor/Receiver) to register as a new user with including the fields such as Username, Blood Group, User City, User Phone Number, User Email and User Password. User must remember the User Email and Password to login into their account.



Rapid Prototype for Hospital Login Feature:

Login – By default one of the security features of this system is the secure login into the system. The login system of this Blood Bank Management System uses a session. It means that the user can only log in at once on the same browser. Hospital User needs to enter the Hospital Email and Hospital Password to login.



Rapid Prototype for Hospital Login Feature:

After successful login into the Hospital User using his/her credentials, below screenshot is the home page for Hospital User. On the Home page we can observe the three lines at the top right corner, which will contain additional features.

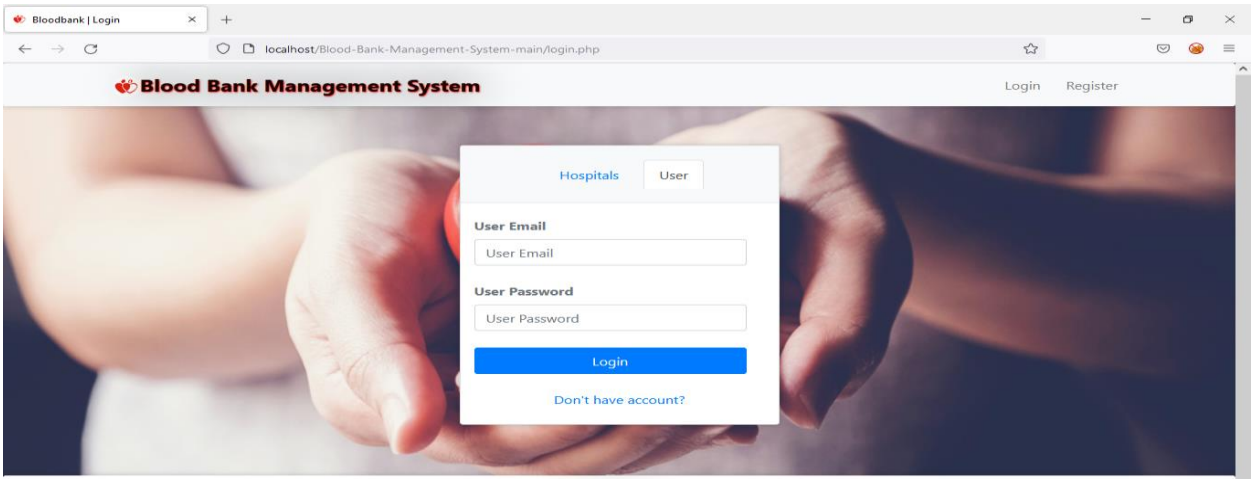
Rapid Prototype for Hospital Login Feature:

Once you click on these three lines, here we can see the features that are only accessed by Hospital User and each feature has its own functionality. The final feature LOGOUT will logout from the account.

Rapid Prototype for User Login Feature:

Login – By default one of the security features of this system is the secure login into the system. The login system of this Blood Bank Management System uses a session. It means

that the user can only log in at once on the same browser. User needs to enter the User Email and User Password to login.

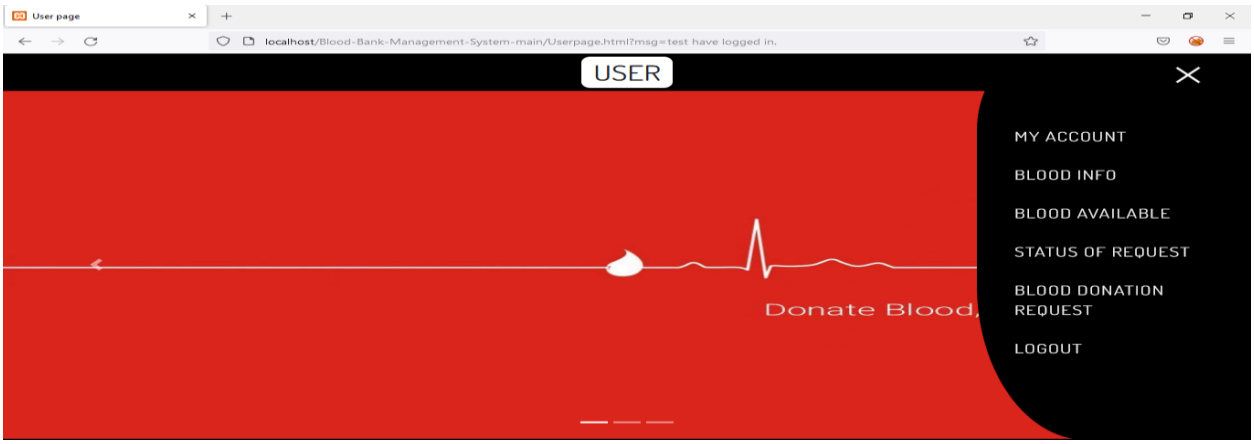


Rapid Prototype for User Login Feature:

After successful login into the User (Donor/Receiver) using his/her credentials, below screenshot is the home page for User. On the Home page we can observe the three lines at the top right corner, which will contain additional features.

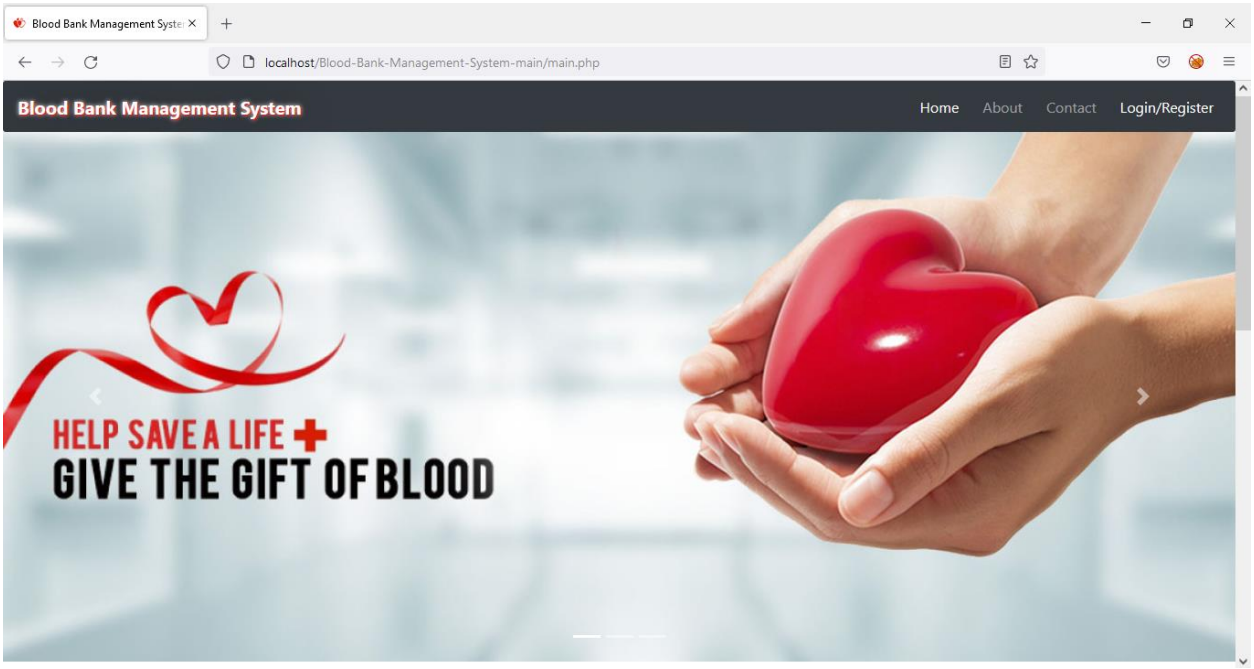
Rapid Prototype for User Login Feature:

Once you click on these three lines, here we can see the features that are only accessed by Hospital User and each feature has its own functionality. The feature STATUS OF REQUEST will give the status of request that you made before to the hospital.



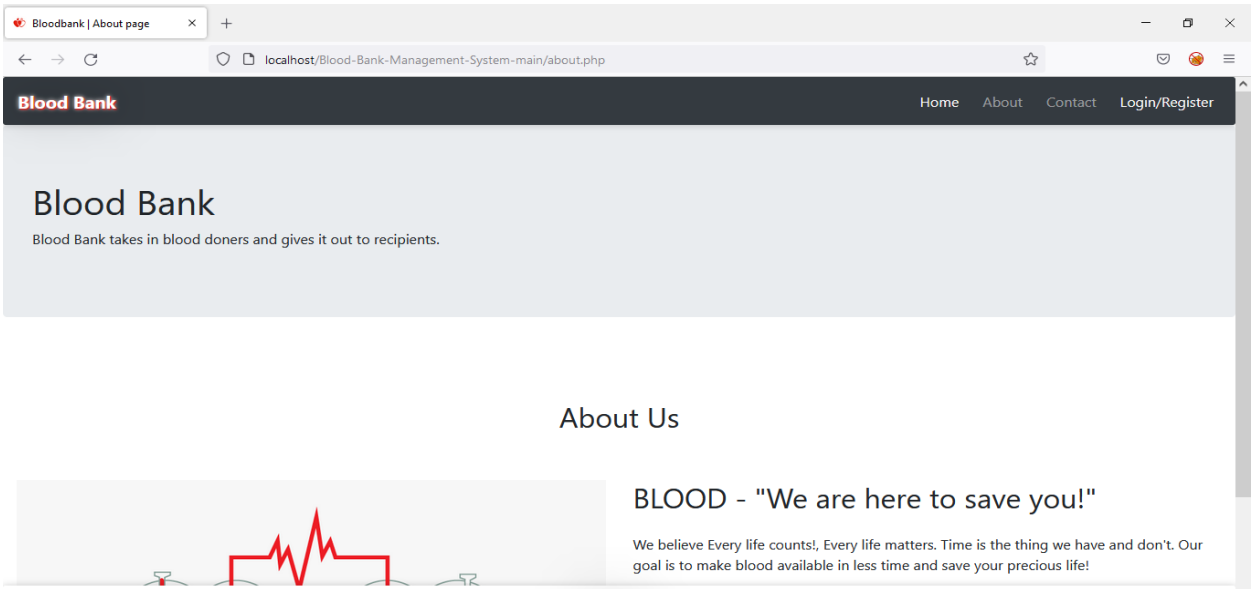
Rapid Prototype for Home Feature:

Whenever any User or Administrator access my software, he/she will see this home page. It has some images scrolling on this home feature.



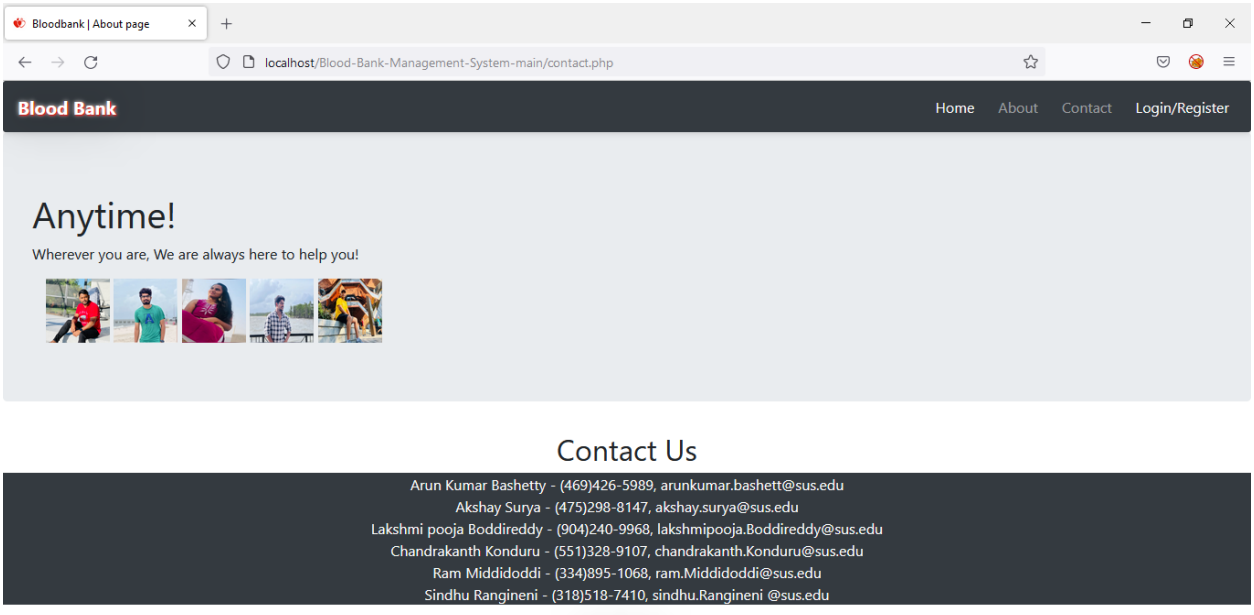
Rapid Prototype for About Feature:

This feature accessible for any user and it has some text which is related to blood.

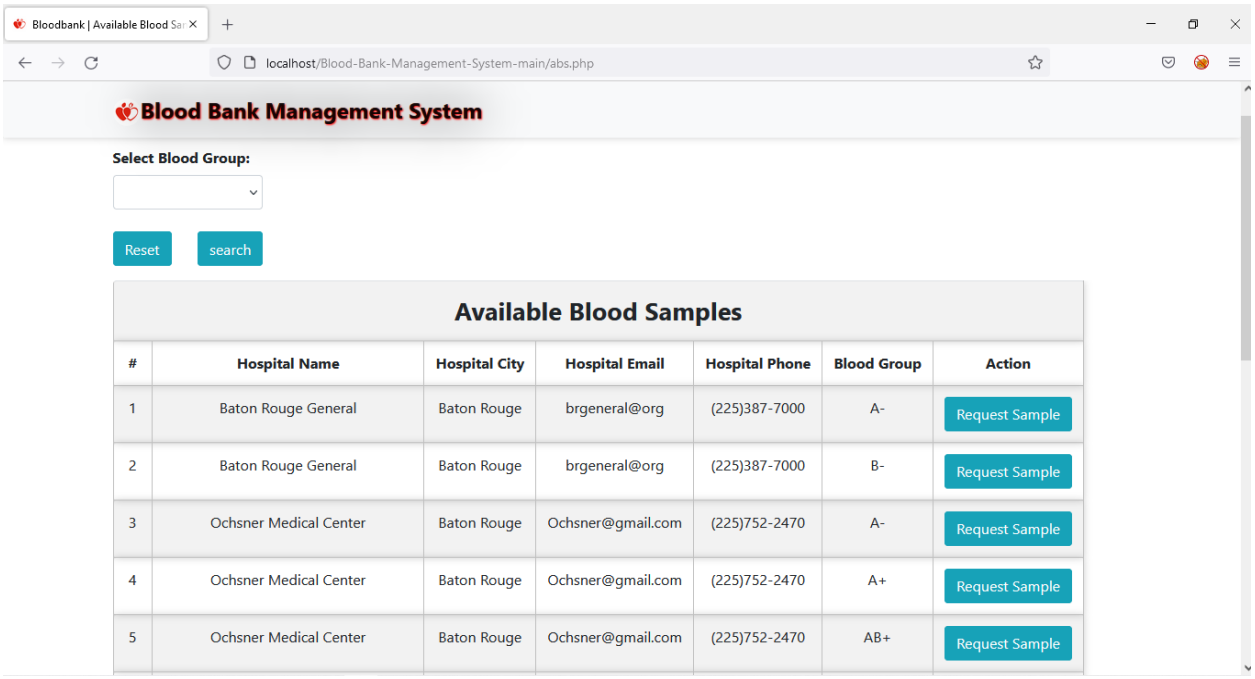


Rapid Prototype for Contact Feature:

Contact Information – For the contact us, you will be able to see our address, phone number and email address as well.



Rapid Prototype for Blood Available Feature:



Testing Protocol:

S.NO	When	Requirement	Verified	Validated
1	**06/09/2022**	Type Of Software	Akshay Surya	Arun BASHETTY
2	**06/09/2022**	Features	Chandrakanth	Ram Middidoddi
3	**06/12/2022**	Platform Requirements	Ram Middidoddi	Sri Laxmi
4	**06/18/2022**	Rapid Prototype	Sindhu Rangineni	Akshay Surya
5	**06/10/2022**	Delimitations	Sri Laxmi	Sindhu Rangineni

DESIGN PHASE

After Requirements, Specifications, Design Phase is the third stage of Software Engineering Life Cycle.

The Design phase is where we look at the many potential solutions to determine the most effective and efficient way to construct the software.

Depending on the project subject, the design phase products include diagrams, flow-charts, HTML Screen designs, prototypes, and UML schemas.

Design phase includes the Class and Sequence diagram for each feature that we have included before in UML Diagram.

In Design phase we have two types of Diagrams,

1. Class Diagram

2. Sequence Diagram

Class Diagram is static, which is Organizing the modules.

Sequence Diagram is Dynamic, which goes into details; it is modern technique, each feature has one sequence.

Goals

Its primary purpose is to transform all the requirements into complete, detailed system design specifications.

Once your design is approved, the Development Team begins its development work.

The list below is the blood bank management system at **Design Phase:**

- Class Diagrams
- Sequence Diagrams
- Testing Protocol

Class Diagram for Registration Feature:

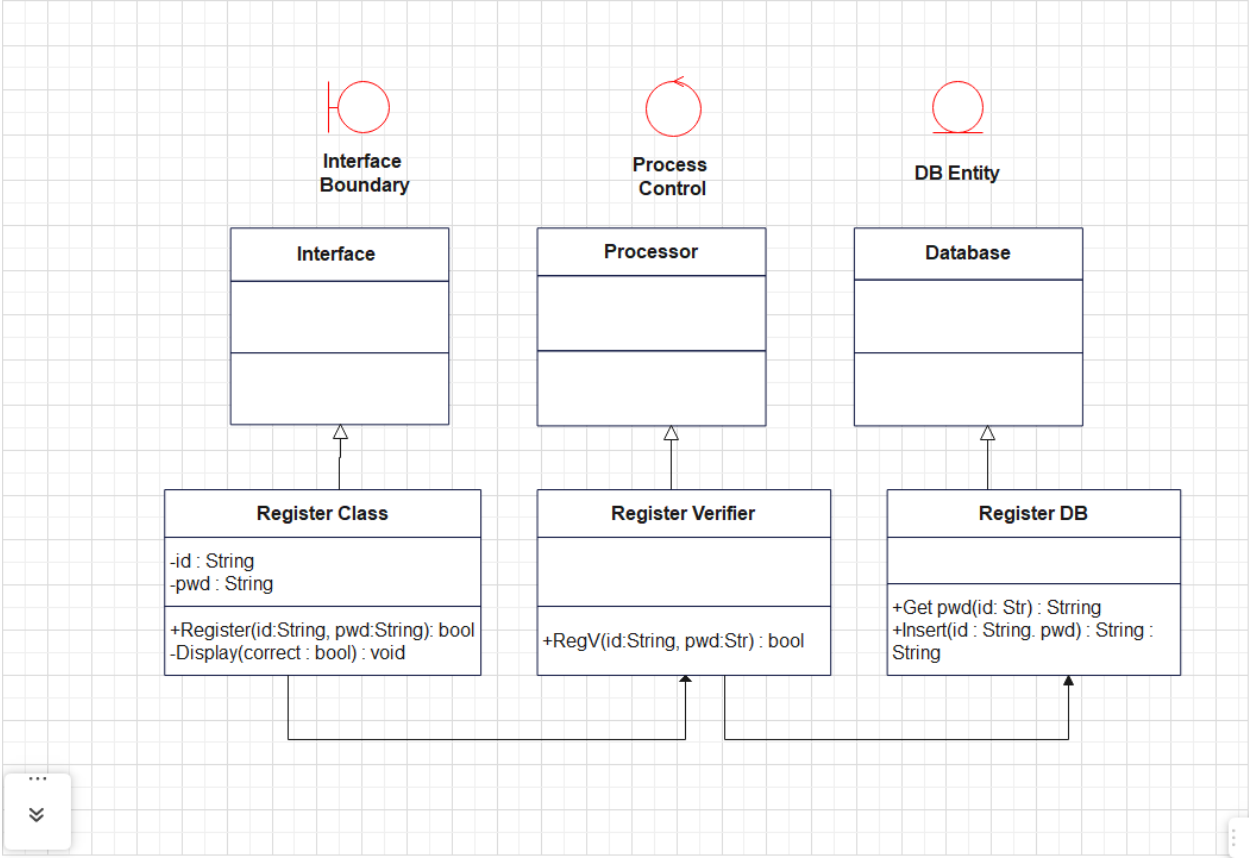
The first feature we have in our software is Registration, this below image show how the web interface communicate with process control and how process control communicates with Database.

If you can observe the below image, we represented the classes as Register class, Process Control as the Register Verifier and, Database as the Register DB.

We represent instance variable as (-) which are Private.

We represent methods as (+) which are public, each class as must Public Class.

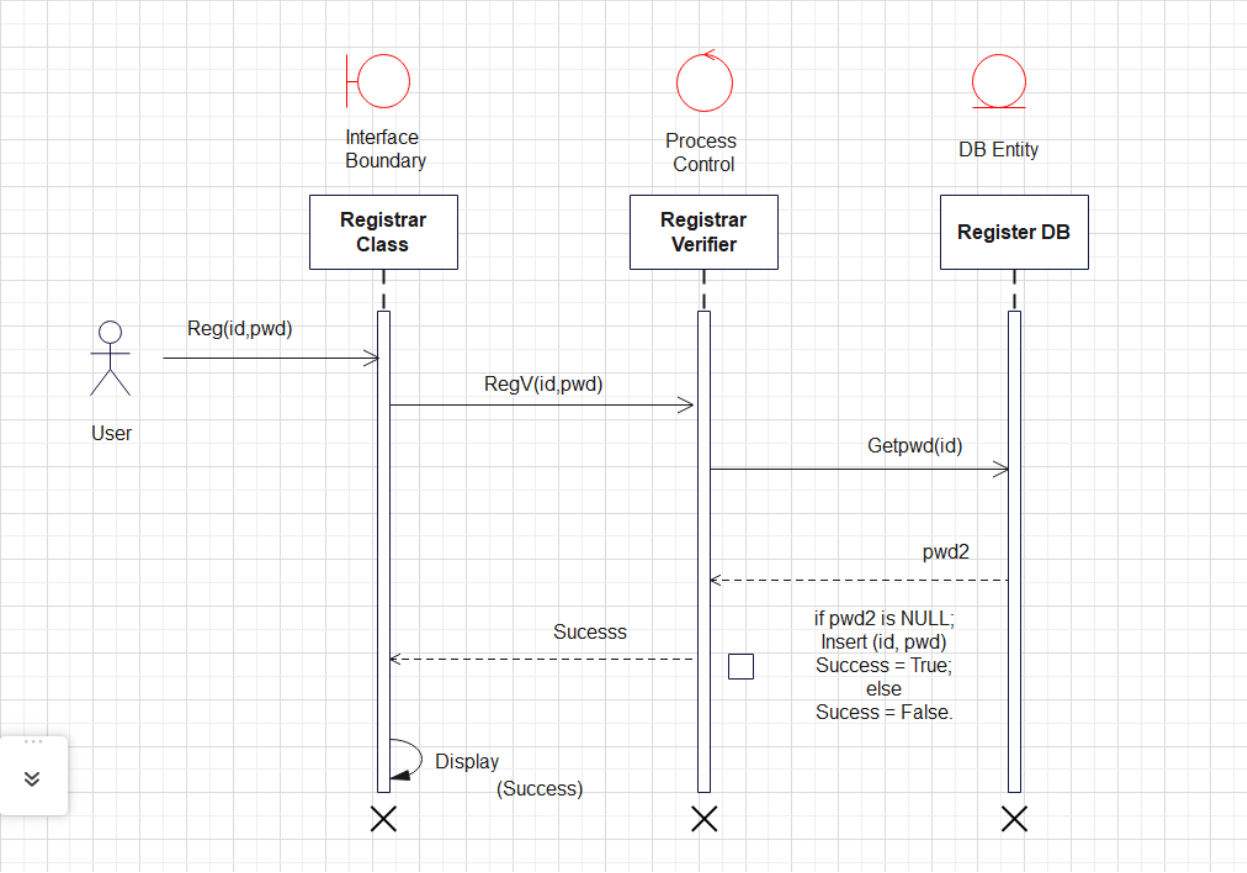
In below class diagram, id, Strings are Private, Display are Private and remaining are public methods such as Register (id: String, pwd: String): bool, RegV (id: String, pwd: String): bool and, Getpwd (id: Str): String.



Sequence Diagram for Registration Feature:

Firstly, the admin user tries to provide id and pwd to register as a new user using web interface, Register Verifier take the id, pwd from interface and ask the Register DB to get the pwd.

In Register DB have two options, if pwd2 is not exists in the Database using insert method (id, pwd) try to store the pwd2 in database and success notification will send to the processor and Display success on Interface.



Class Diagram for Login Feature:

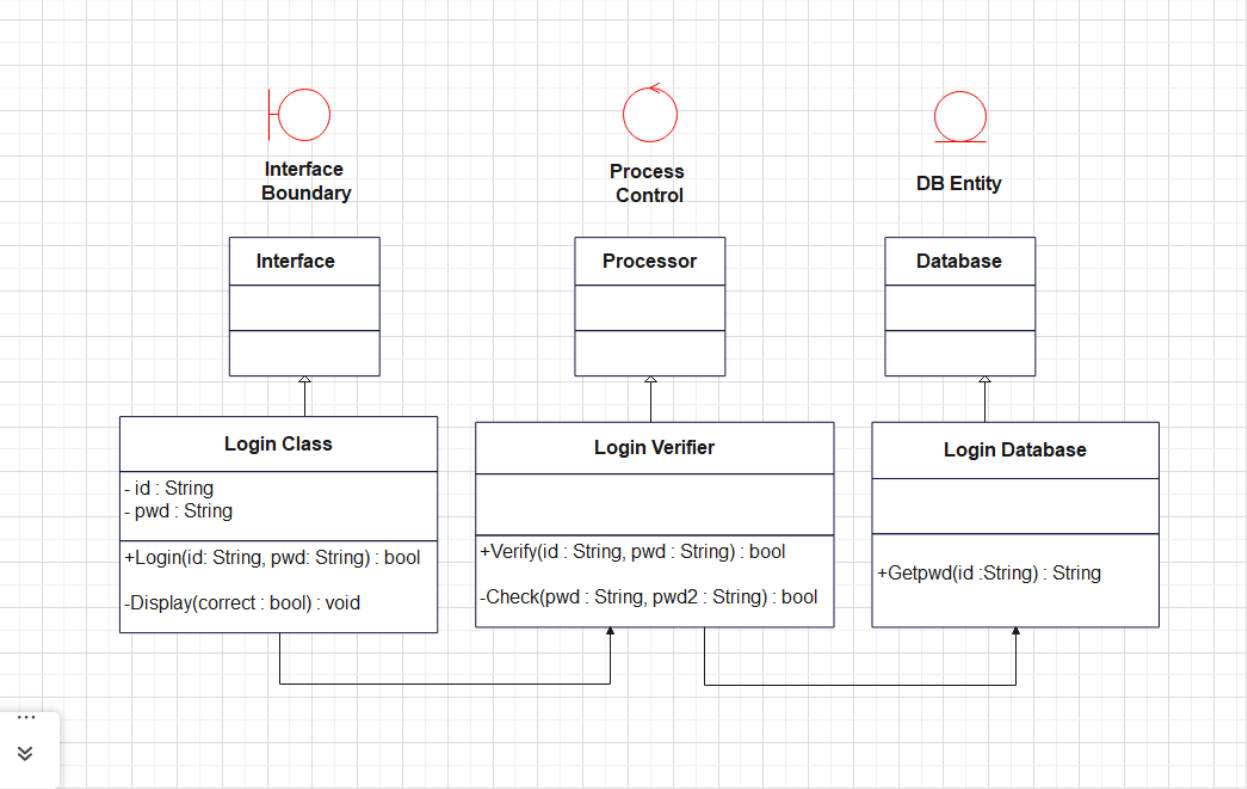
The second feature we have in our software is Login, this below image show how the web interface communicate with process control and how process control communicates with Database and, also will see how database returns the response to process control, displays information on the web interface.

If you can observe the below image, we represented the class interface as Login class, Process Control as the Login Verifier and, Database as the Login DB.

We represent instance variable as (-) which are Private.

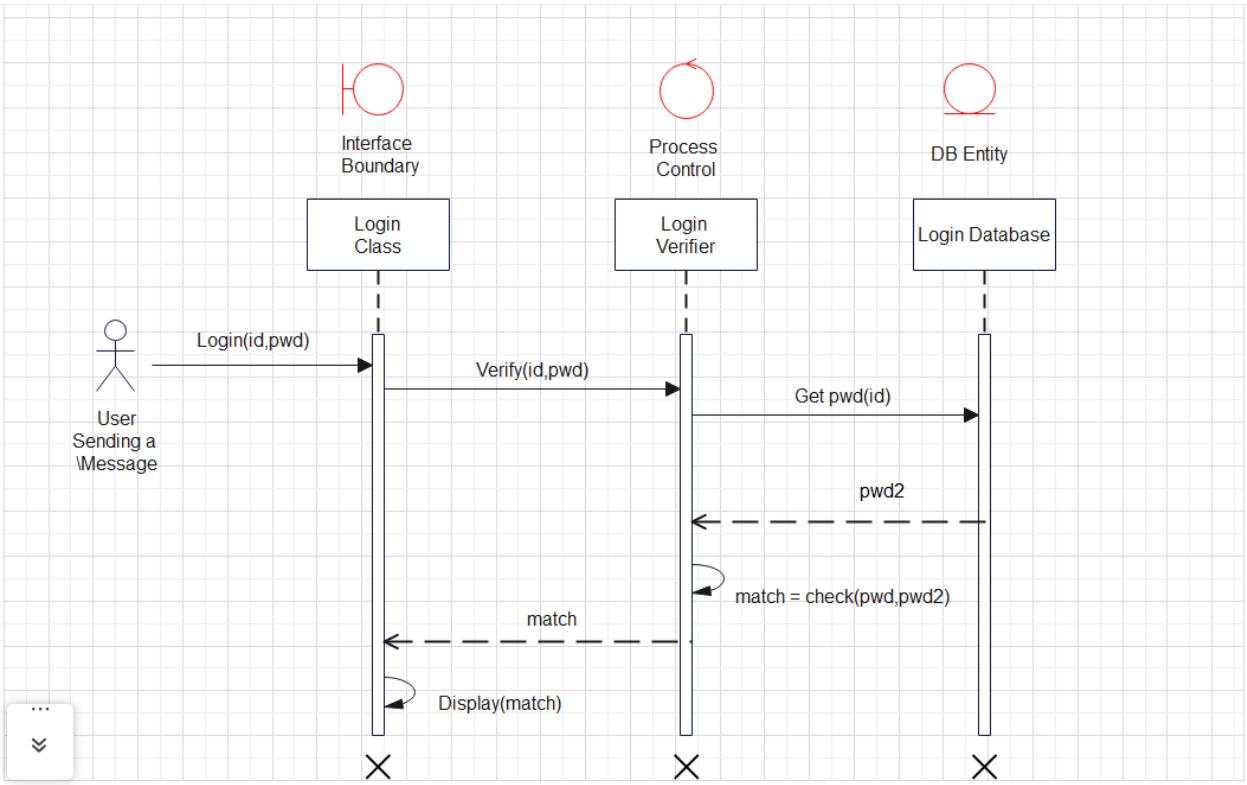
We represent methods as (+) which are public, each class as must Public Class.

In below class diagram, id, Strings are Private, Display are Private and remaining are public methods such as Login (id: String, pwd: String): bool, Verify (id: String, pwd: String): bool and, Check (pwd: String, pwd2: String): bool, Getpwd (id: String): String.



Sequence Diagram for Login Feature:

The User send a request to login into the software using the Login Id and password, Login verifier try to verify the credentials provided by the User by communicating with Login Database and Database will check the password with saved password (pwd2), once verification is completed, if the password is match , on the interface we can see the login successfully notification.



Class Diagram for Home Feature:

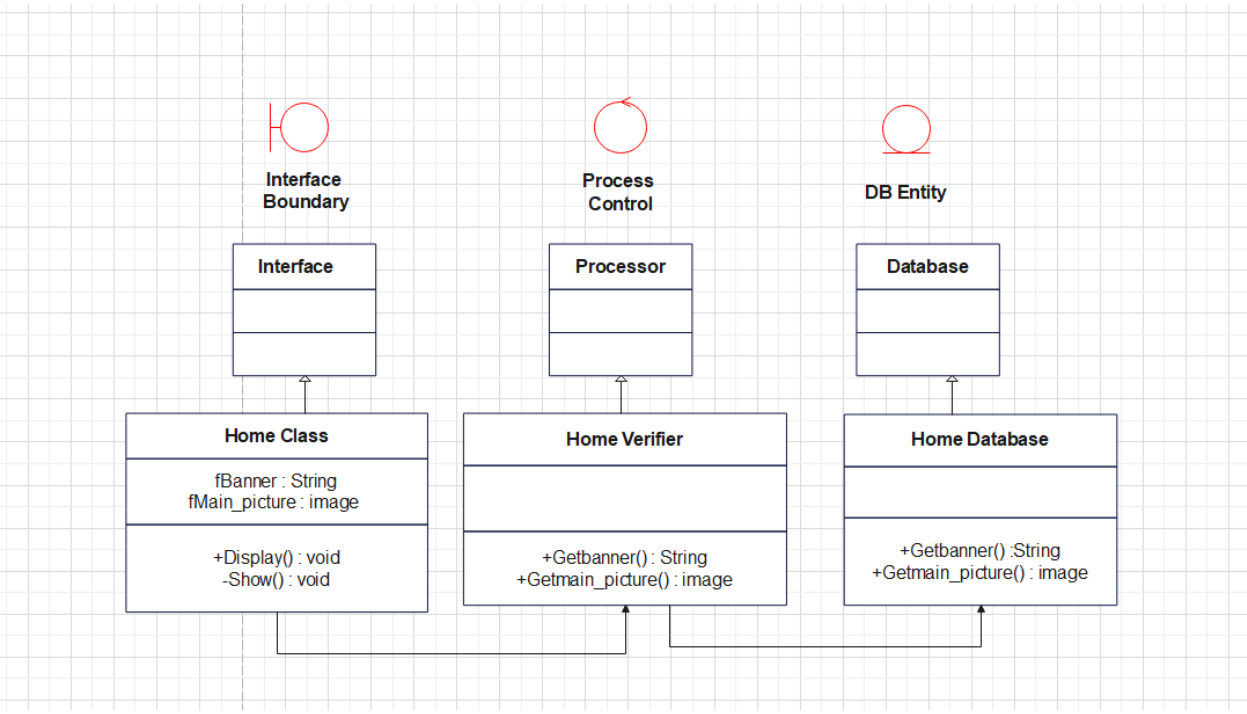
The third feature we have in our software is Home, this below image show how the web interface communicate with process control and how process control communicates with Database and, also will see how database returns the response to process control, displays information on the web interface.

If you can observe the below image, we represented the class interface as home class, Process Control as the Home Verifier and, Database as the Home Database.

We represent instance variable as (-) which are Private.

We represent methods as (+) which are public, each class as must Public Class.

In the below class diagram, I have used instance variables such as fBanner and fMain_picture and those are reference to the String and Image. Show (): void is Private here and remaining are public methods such as Display (): void, GetBanner (): String, Getmain_picture (): image.

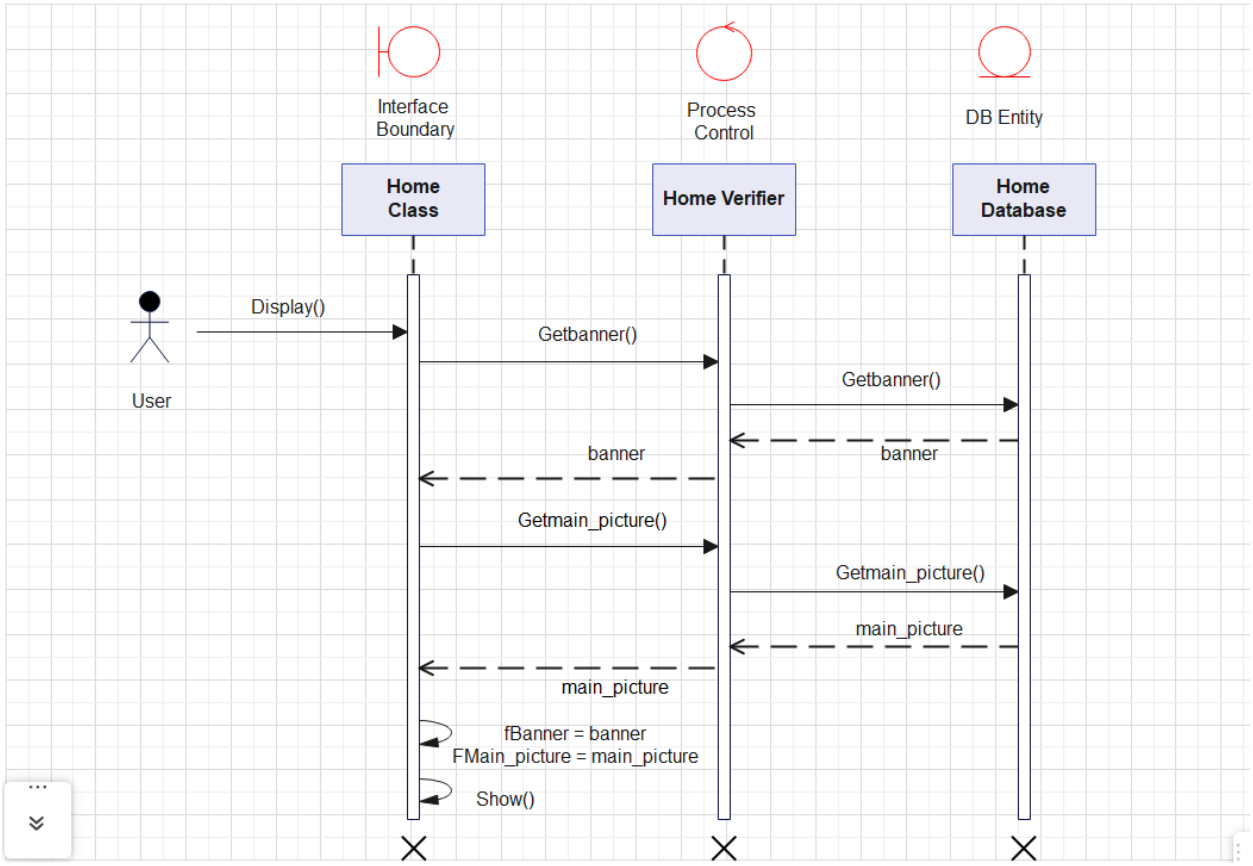


Sequence Diagram for Home Feature:

Sequence diagram for home feature is different from above registration and Login features,

Users want to Display the Home page on the web page and Interface communicate with home verifier to get the context. The text is always stored in Home Database, Home verifier try to communicate with home database to get the data using the method GetBanner () and stored. This same process happens to get the picture from Database using Getmain_picture () method and stored.

Now we used Instance variable to Display the banner and picture on the web page using fBanner and fMain_picture.



Class Diagram for Contact Feature:

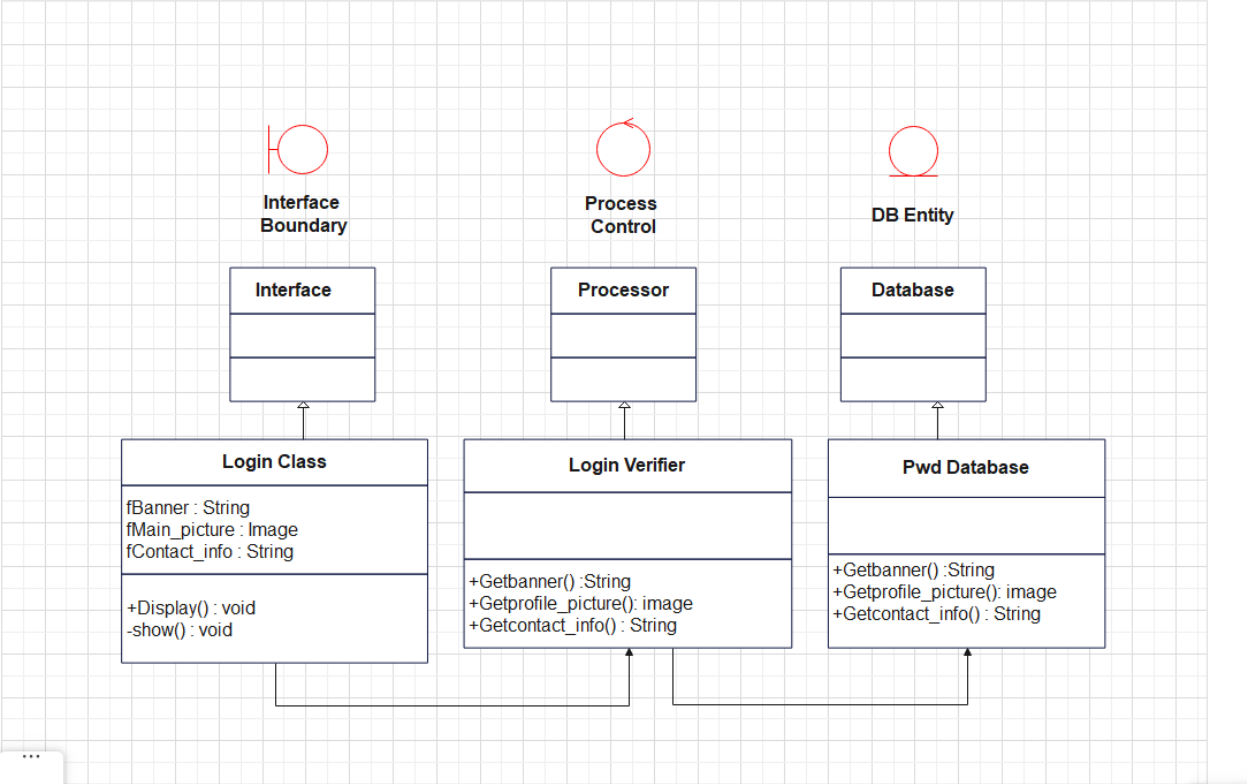
The fourth feature we have in our software is Contact, this below image show how the web interface communicate with process control and how process control communicates with Database and, also will see how database returns the response to process control, displays information on the web interface.

If you can observe the below image, we represented the class interface as Contact class, Process Control as the Contact Verifier and, Database as the Contact Database.

We represent instance variable as (-) which are Private.

We represent methods as (+) which are public, each class as must Public Class.

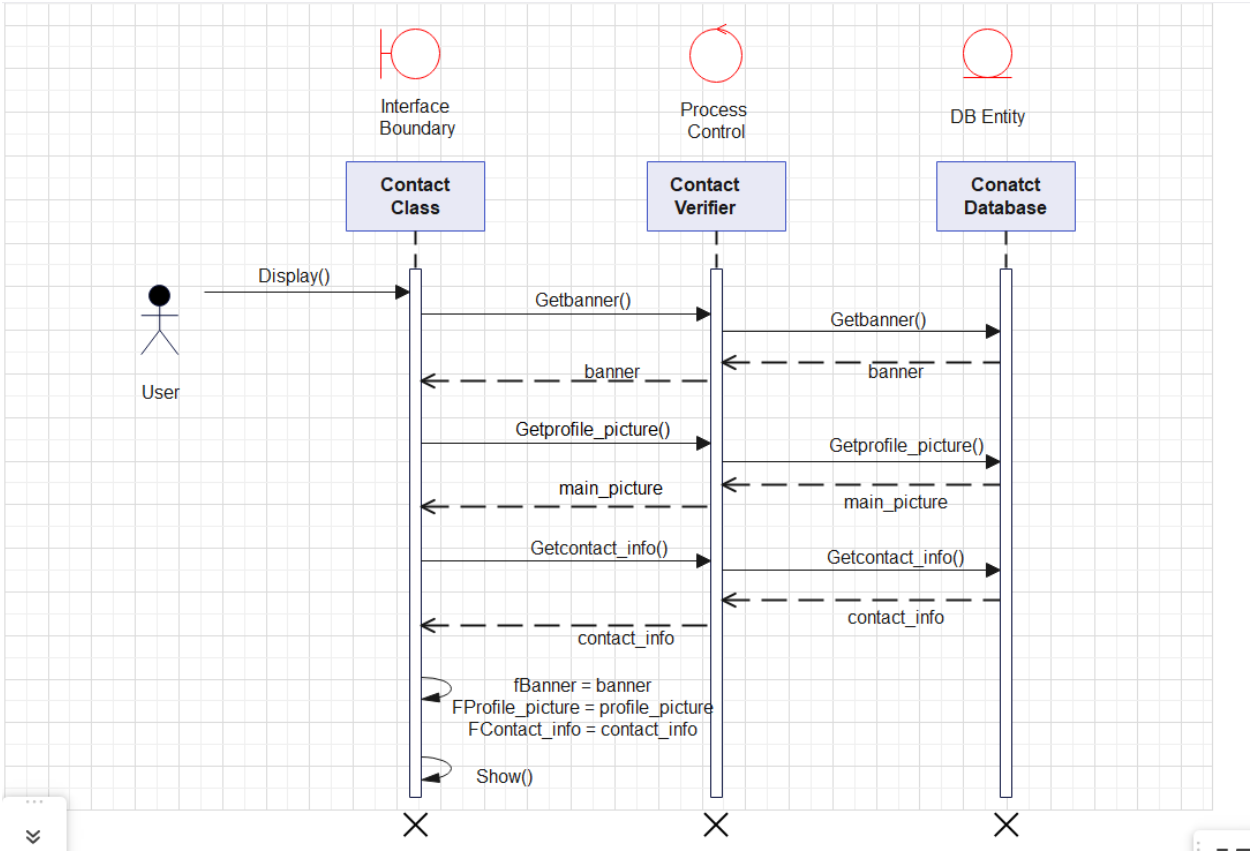
In the below class diagram, I have used instance variables such as fBanner and fMain_picture same as home class diagram but extra included fContact_info: String and, those are reference to the String, Image, and String. Show (): void is Private here and remaining are public methods such as Display (): void, GetBanner (): String, Getmain_picture (): image, GetContact_info ().



Sequence Diagram for Contact Feature:

Contact Sequence diagram is like the home sequence diagram, User want to Display the Contact page on the web page and Interface communicate with Contact verifier to get the context. The text is always stored in Contact Database, contact verifier try to communicate with Contact database to get the data using the method GetBanner () and stored. Same process happens to get the picture from Database using Getmain_picture () method and stored.

In this below sequence diagram, we included extra method called Getcontact_info (), this method is used to get the contact information from Database. Once we completed the to get the banner, image and, contact information, using instance variables fBanner, fProfile_picture, fContact_info we displayed all the information on web page.

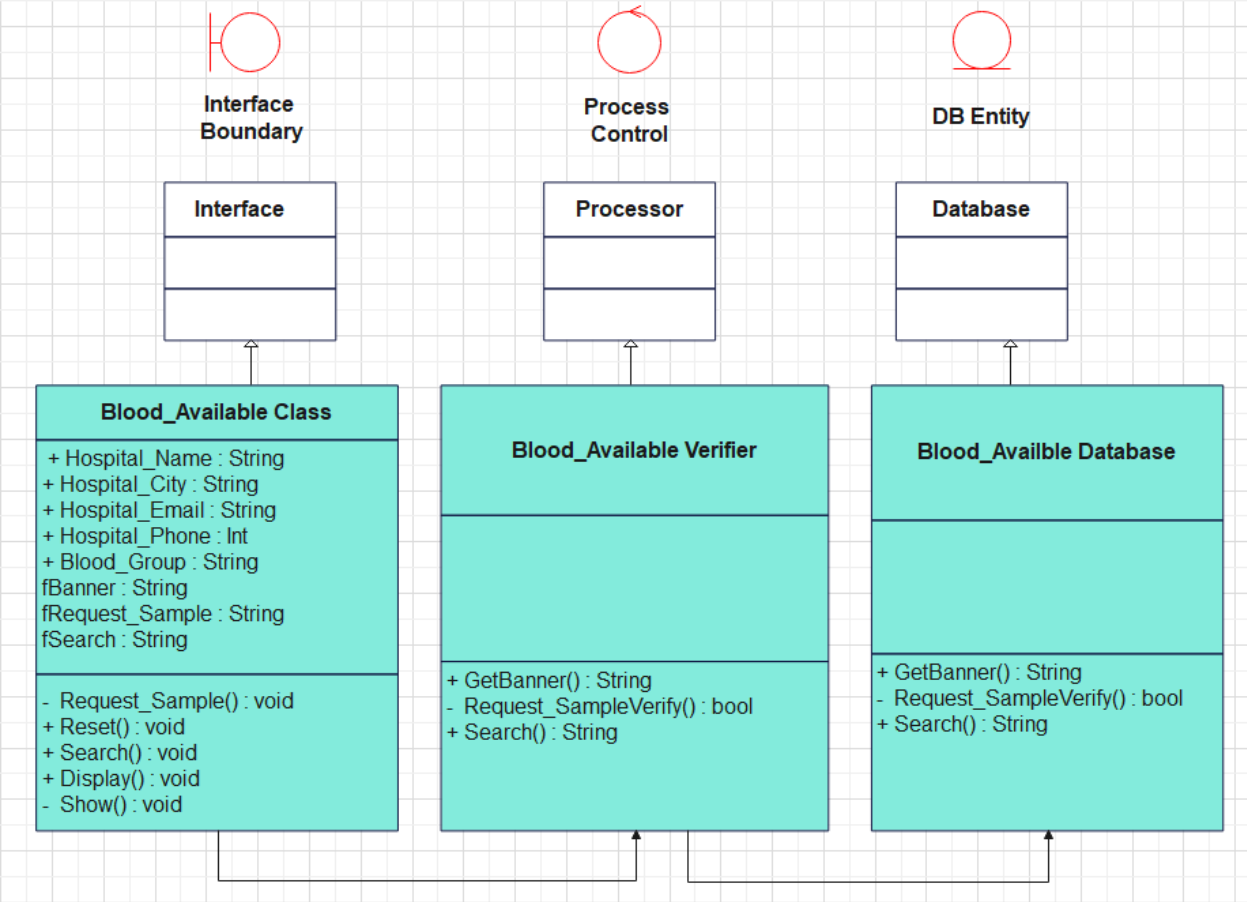


Class Diagram for Blood Available Feature:

The fifth feature we have in our software is Blood Available, the below class diagram represents when user want to display the type of blood available in Hospitals and this feature is only accessed by the Hospital Admin User.

The Blood_Available class contains five attributes which are public and, related to the Hospital and, we have few operations also included such as Request_Sample, Show are Private.

As same as Contact class diagram, we have Banner public method which will used to get the text from database. Finally Public Search method used to search the information.

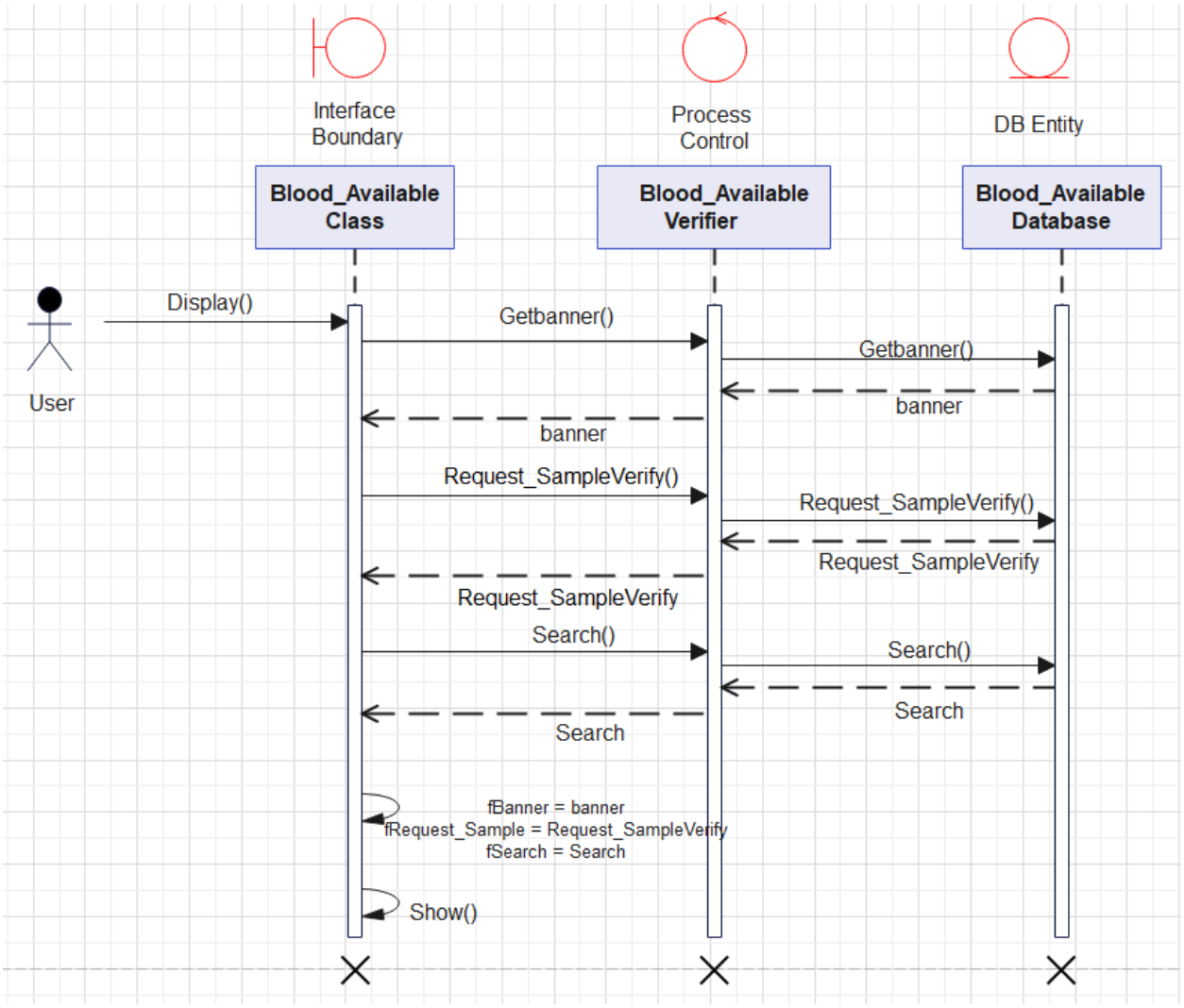


Sequence Diagram for Blood Available Feature:

Users want to get the Request_Sample which has shown on the Blood_Available web interface, for this request, first we used Getbanner () method to get the text from the database.

We used Request_SampleVerify () method to Request blood group to the hospital, when user try to request sample, processor will send notification to the all the hospitals stored in the database.

Search () method is public, and it is used to search blood group which user wants. Finally, instance variables such as banner, Request_Sample, Search are used here to display all the information on web interface.


Interface Boundary
Process Control
DB Entity
Blood_Available Class
Blood_Available Verifier
Blood_Available Database
User
Display()
Getbanner()
Getbanner()
banner
banner
Request_SampleVerify()
Request_SampleVerify()
Request_SampleVerify
Request_SampleVerify
Search()
Search()
Search
Search
fBanner = banner
fRequest_Sample = Request_SampleVerify
fSearch = Search
Show()

Testing Protocol

S. No	When	Which Feature Tested	Who Verified	Who validated
1	07/08	Register Class	Akshay Surya	Arun BASHETTY
2	07/08	Register Sequence	Ram Middidoddi	Lakshmi Pooja
3	07/09	Login Class	Chandrakanth	Venkat Gummadidala
4	07/09	Login Sequence	Sindhu Rangineni	Ram Middidoddi
5	07/09	Home Class	Lakshmi Pooja	Sri Laxmi Miryala
6	07/10	Home Sequence	Sri Laxmi Miryala	Sindhu Rangineni
7	07/12	Contact Class	Akshay Surya	Lakshmi Pooja
8	07/12	Contact Sequence	Venkat Gummadidala	Chandrakanth
9	07/15	Blood Available Class	Arun BASHETTY	Akshay Surya
10	07/15	Blood Available Sequence	Ram Middidoddi	Arun BASHETTY

Detailing the activity and corrections for Testing Protocol,

Register Class:

Class diagram for register feature is verified **Akshay Surya**, it was the first feature, and he took longer time to understand how the class diagram work and after he finished verifying the class diagram, I (**Arun BASHETTY**) did some modifications and validated it.

Register Sequence:

Sequence Diagram for register feature is verified by **Ram Middidoddi**, it was the first feature same as class diagram for Register feature, he knew how the public and private methods work, he produced a solution to use Getpwd method and finally Laxmi Pooja validates it.

Login Class:

Class diagram for Login feature is verified by **Chandrakanth,** Login Class diagram is same as the registration class diagram and just added the public method Verify and Private method Check in Processor, Venkat validates it.

Login Sequence:

Sequence diagram for Login feature is verified by **Sindhu Rangineni**, which is same as registration sequence diagram, we did not have any issues in this stage and checked the matches between pwd, pwd2, at the end it is validated by the Ram Middidoddi.

Home Class:

Class diagram for home feature is verified by **Laxmi Pooja**, here she produced a solution of using instance variable such as fBanner, fMain_picture. We used this method in all the sequence diagrams, finally the Sri Laxmi Miryala validate it.

Home Sequence:

Sequence diagram for home feature is verified by **Sri Laxmi Miryala**, we had some issues with using the arrows such as Navigability and Interface, as a team we work together and find the solution to use proper arrows, at the last the Sindhu Rangineni validates it.

Contact Class:

Class diagram for Contact feature is verified by **Akshay Surya**, this class diagram is same as the home class diagram and after reviewing the web interface from Rapid prototype, Akshay got a thought to include the new instance variable fContact_info () and it is validated by Laxmi Pooja.

Contact Sequence:

Sequence diagram for Contact feature is verified by **Venkat,** it is same as home sequence diagram, based upon the class diagram for Contact he tried to add the navigability and Interface arrows, few of them are wrong and Chandrakanth made some error corrections and validated.

Implementation Phase

In implementation Phase we have User manual and Source code,

USER MANUAL:

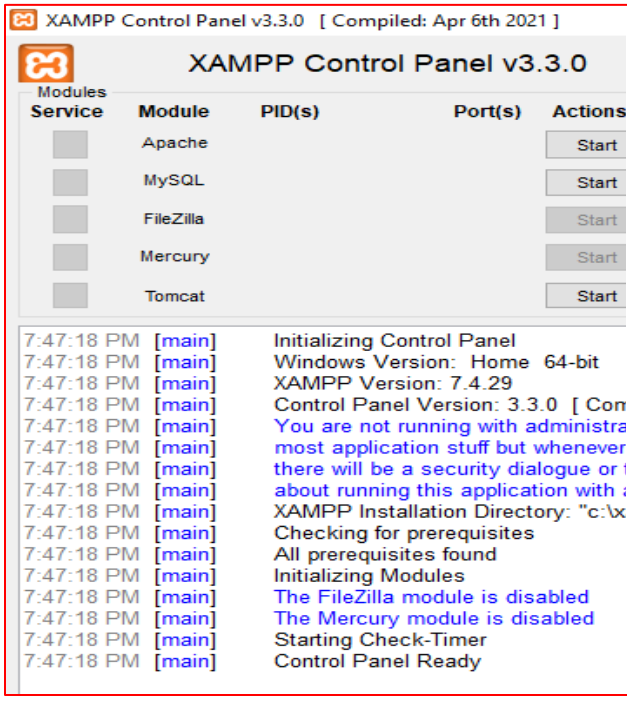
In our project as we mentioned earlier, we have two types of users which are Hospital User and Donor/Receiver. Hospital User has administrative access, and he has the rights to check all the available features and, also for the Donor/Receiver they have their own rights to access all the features.

Languages/Technologies used,

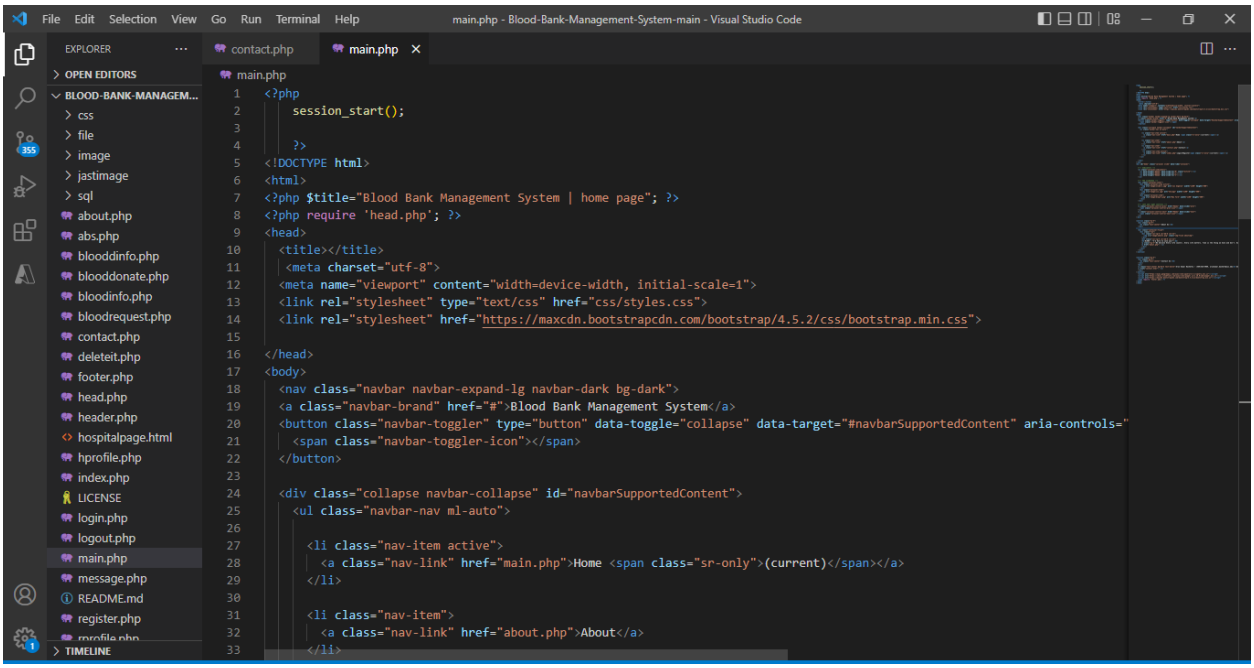
1. Frontend: HTML, CSS, Bootstrap, JavaScript
2. Backend: Database – MySQL
3. Frontend – Backend Connection: PHP

Requirements:

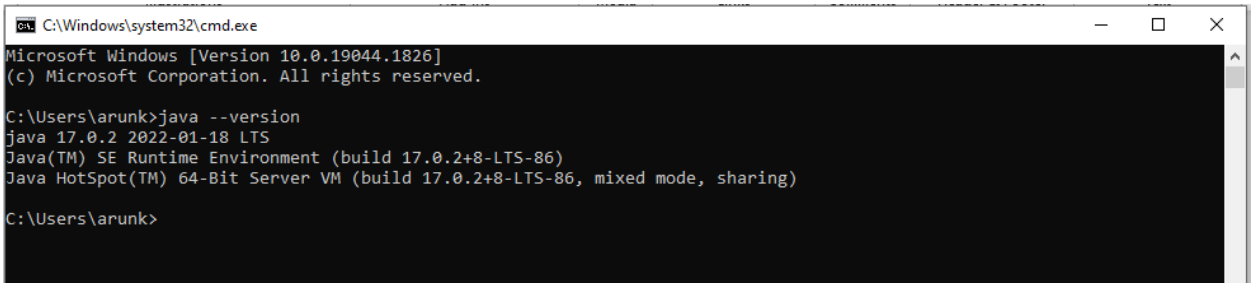
1. **Xampp Software**



2. Visual Studio Code software for supporting php, html and, css



3. Java jdk



How to start the Xampp server and run this software in local system

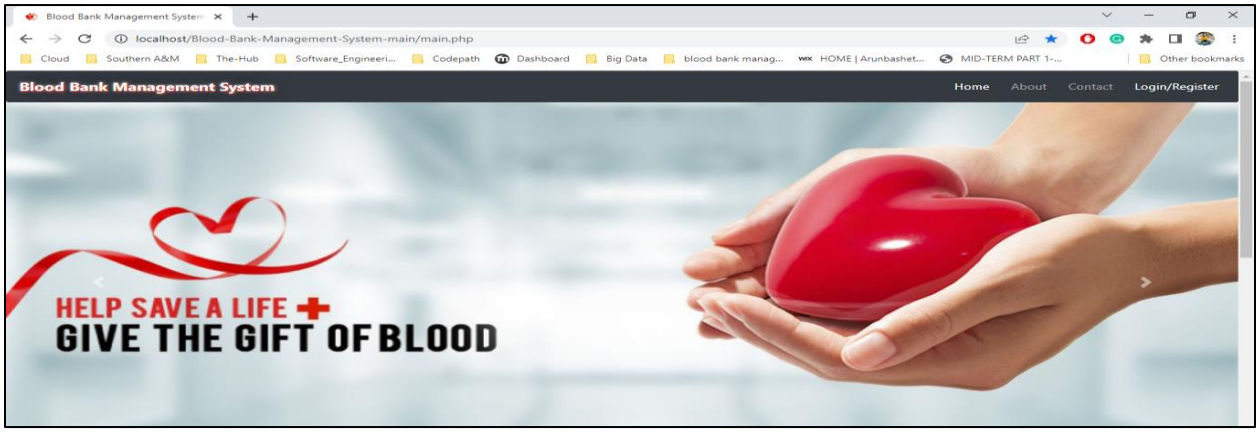
- Download this code as a zip file into your local system.
- Place this entire folder in htdocs, in xampp (xampp path, installed as per your installation).
- Open Xampp server, start Apache, MySQL.
- In the MySQL row, select Admin.
- Create a database with name "blood bank" in phpMyAdmin. Import the sql file from sql folder.

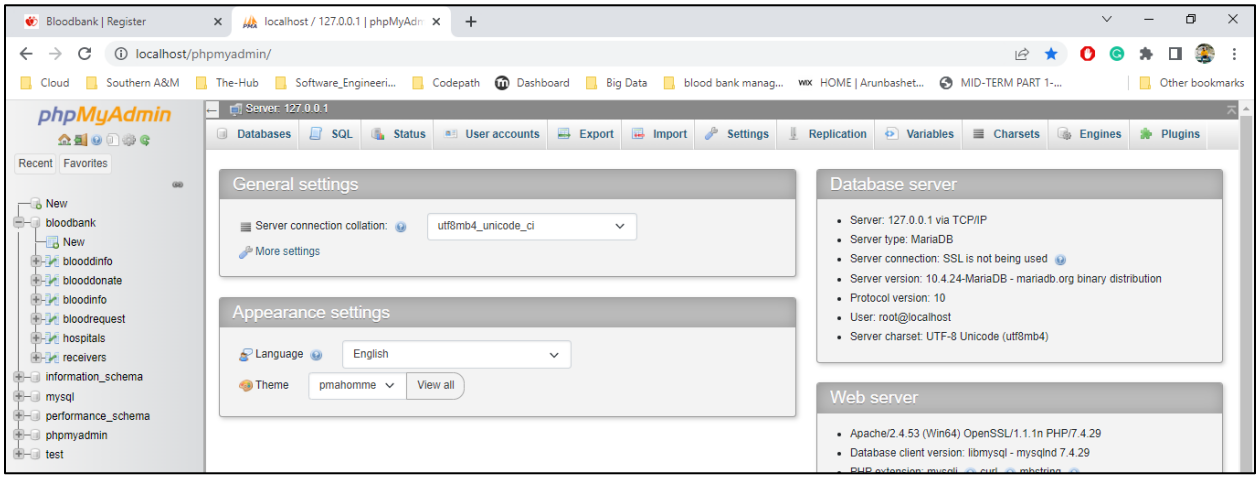
- Open main.php in Sublime text/Visual studio code, on right click copy file path.
- Paste it on any web browser and clear everything before folder name, type "localhost".
- Ready to go!

I have four types of files in our software,

- css' contains the files for the front-end pages style.
- 'file' contains the files for the backend connections.
- 'image' and 'jastimage' contains images used in the front end.
- 'sql' contains the database file that must be imported.

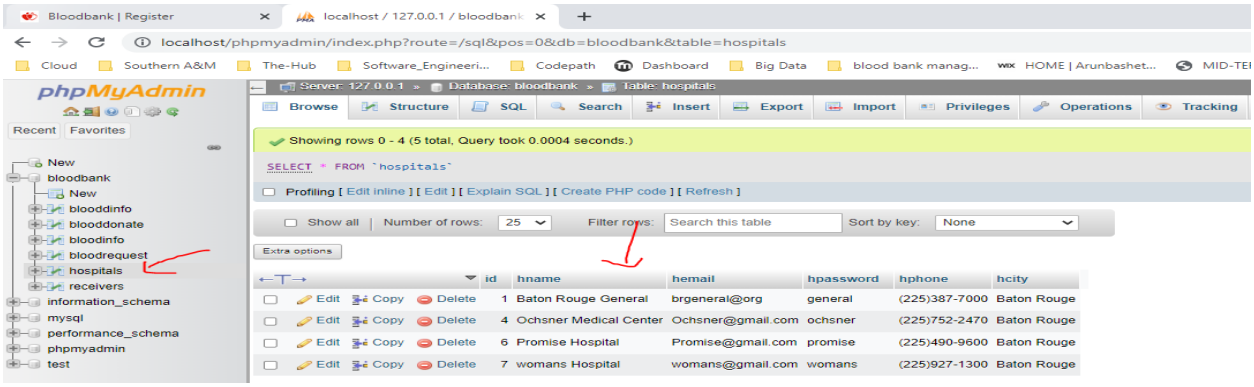
This is the main web page we could see if we successfully installed required applications and run the Apache and phpadmin in xampp server and PHP admin page.



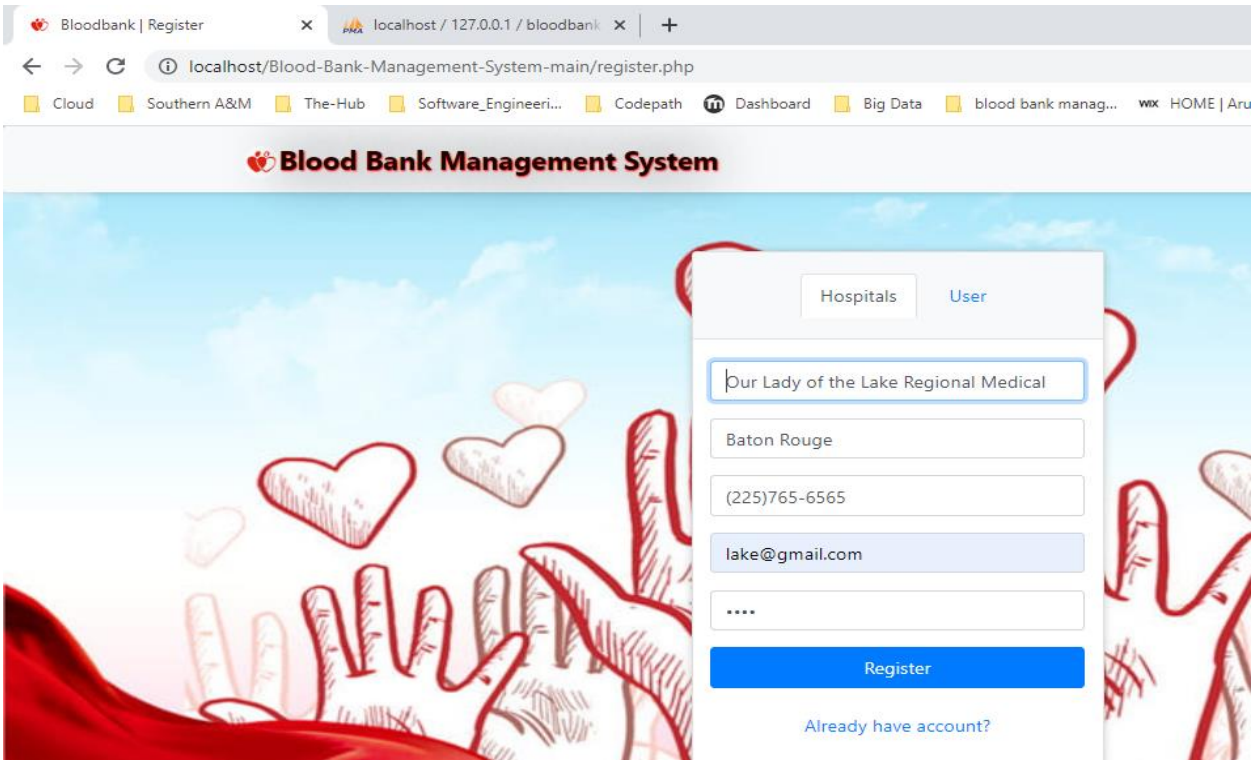


Our main goal of project is as a hospital user or Donor/Receiver first they must register as a new user to access the application for that they need to click on Login/Register from the main web page. Before entering all the information about the hospital this is the php admin.

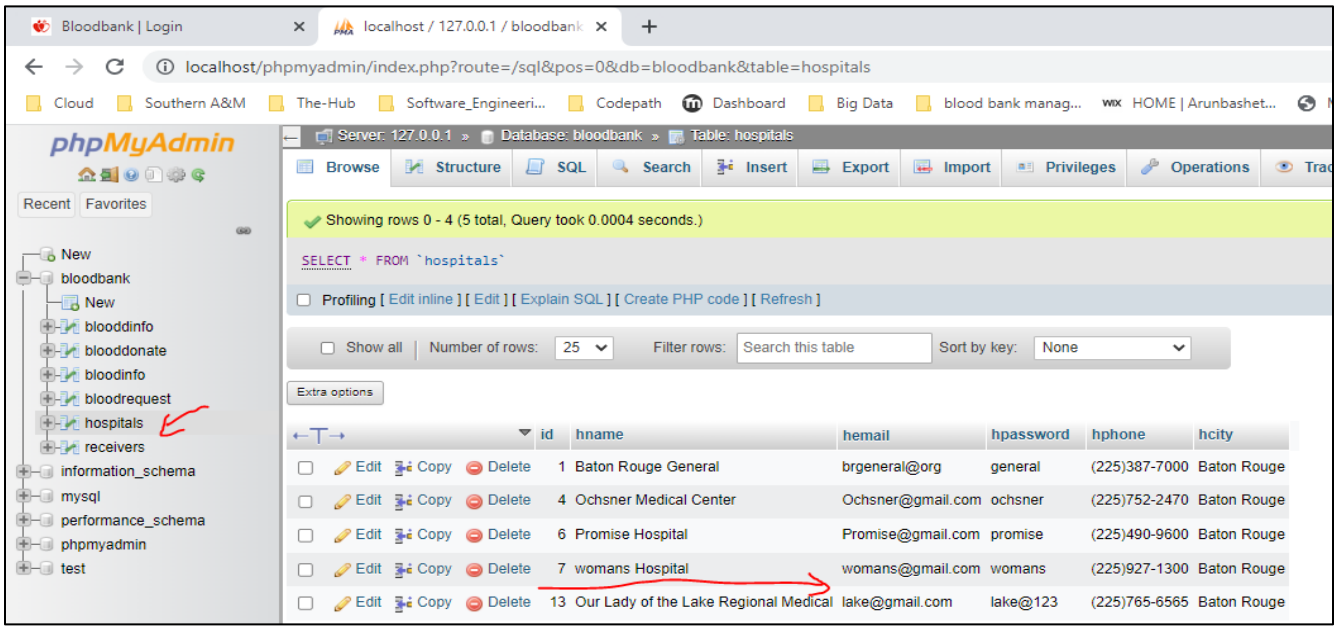
Registration as a Hospital User and Login:



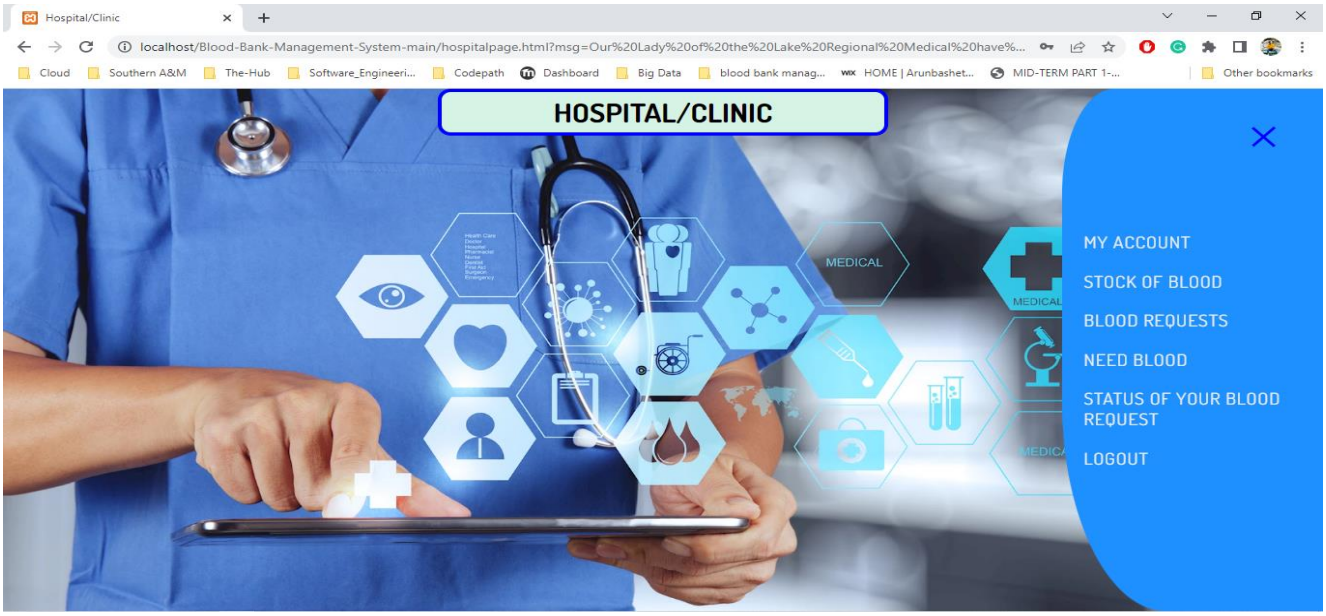
It will ask to enter all the details about the hospital such as Hospital name, Hospital City, Hospital Phone Number, Hospital Email and Hospital Password, I entered this below information and clicked on register. Once all the details are correctly entered you can see the notification on the top showing You have successfully registered. Please, login to continue.



After adding the new Hospital information, on the php admin page you can see new column included in hospital field.

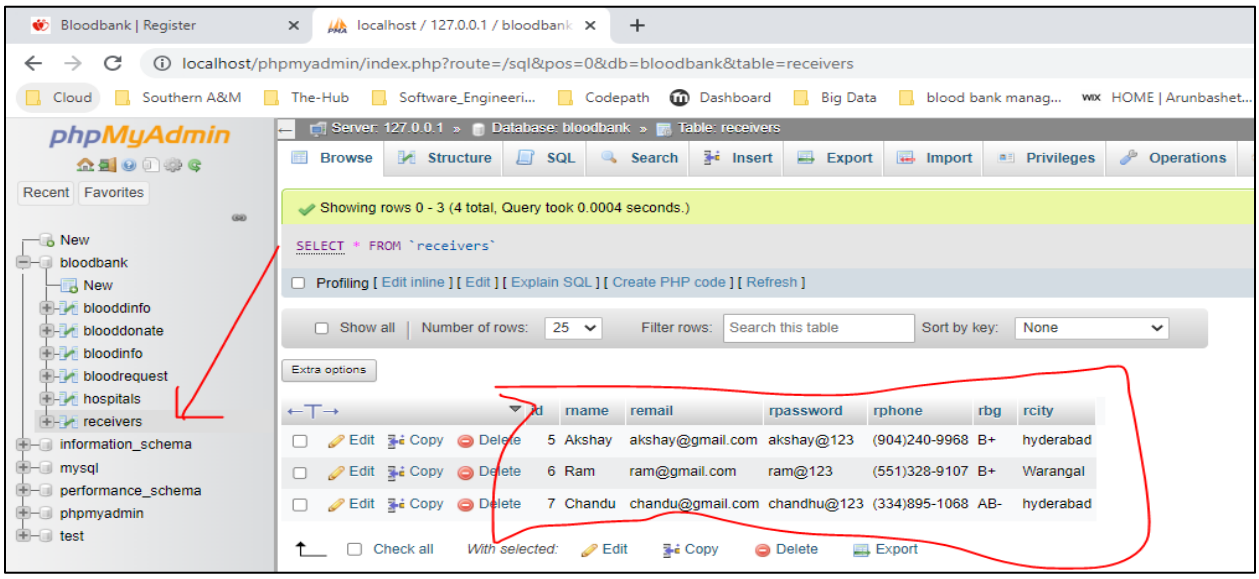


I used new Hospital user credentials to login and here the web page for Lady of the Lake Hospital, you can see the features on the right side.

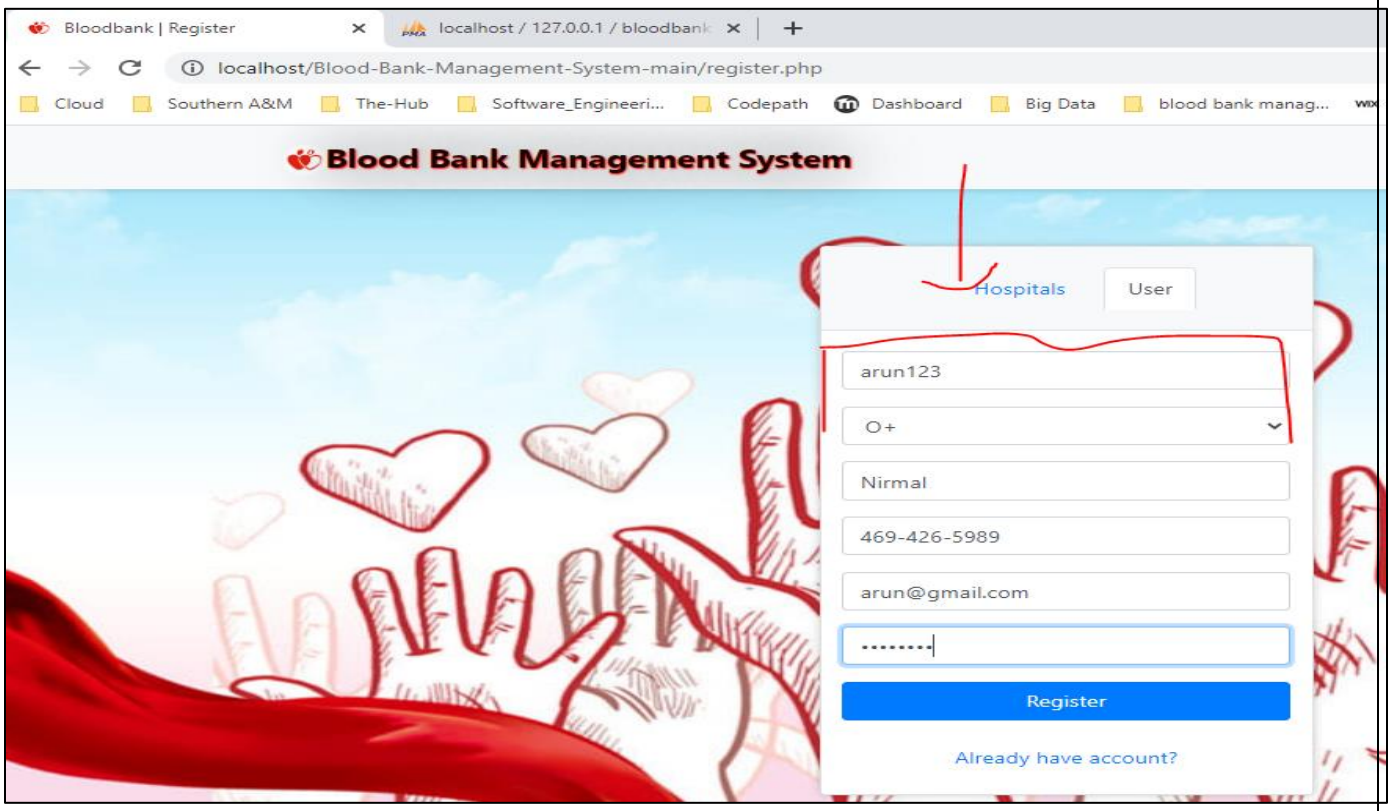


Registration as a Donor/Receiver User and Login:

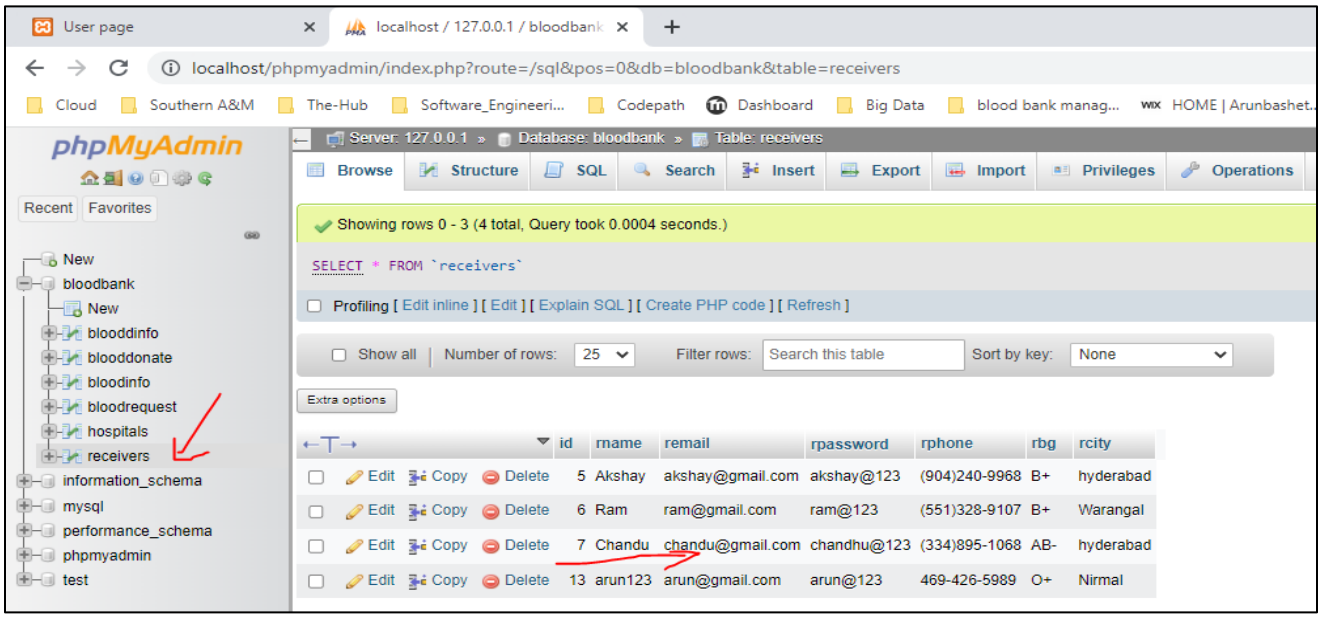
On the php admin page, if you observer there are only three users,



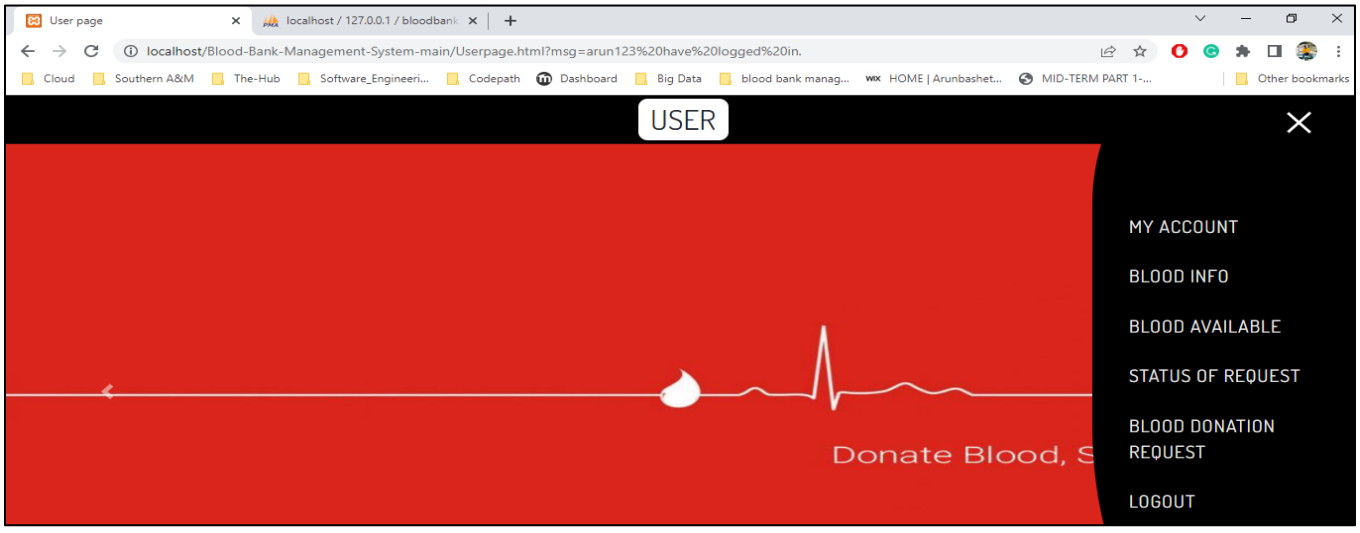
Now I will add one newer user with provide all the information in this below,



After adding the new user called Arun, on the database php admin you can be able to see the new user in receiver's field.

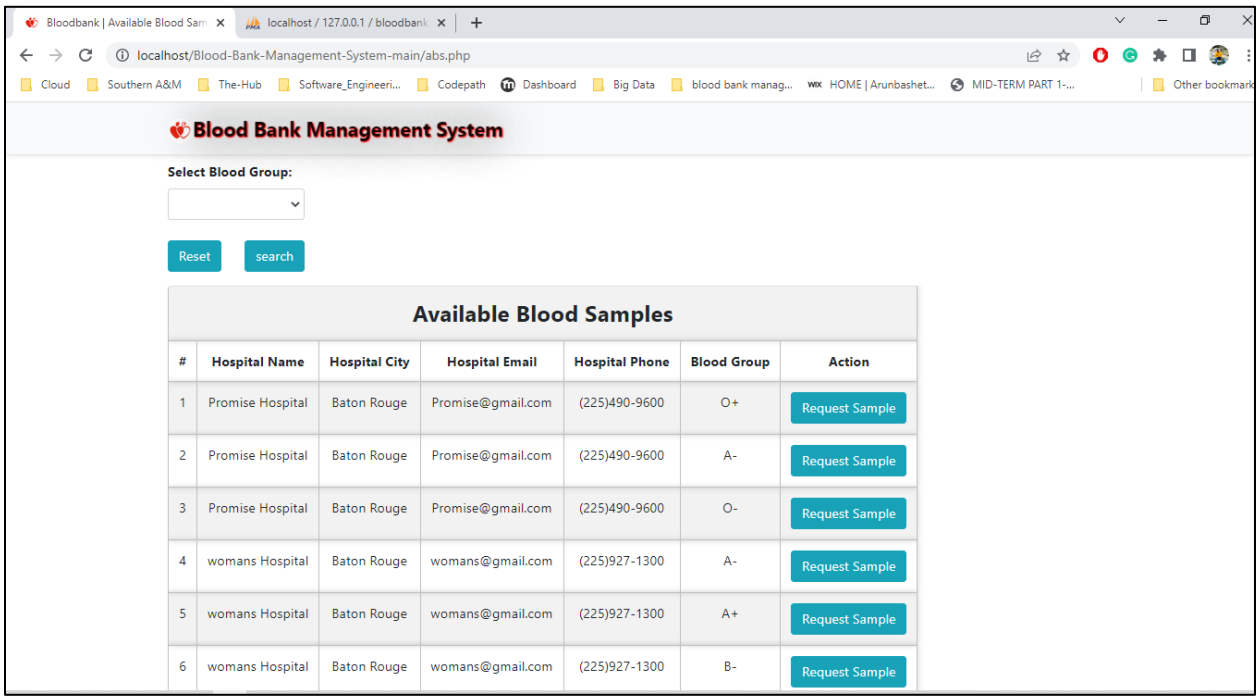


After in login page, I used new user credentials and logged in, here is the login page for Donor/Receiver web page and, also you can see the features which are accessible by user as a admin.

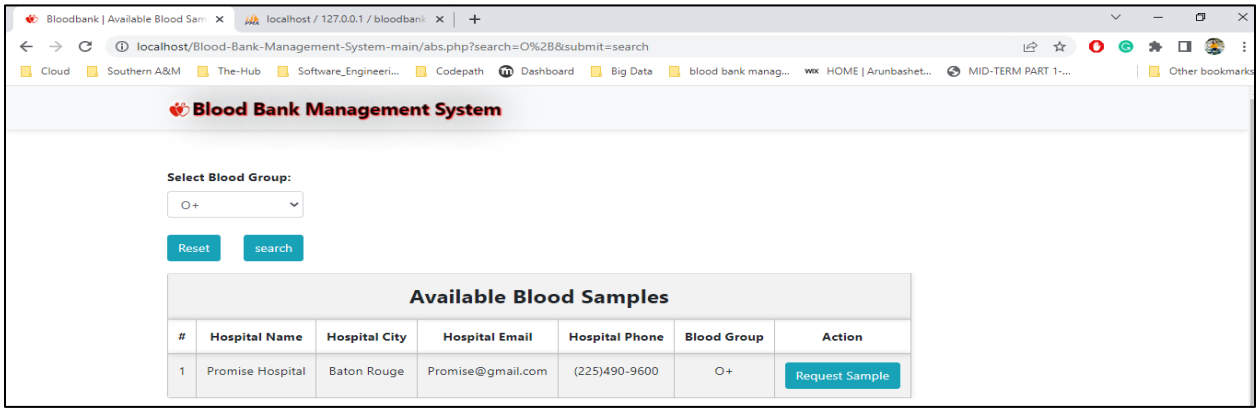


Now the main goal of this software is, Arun user has some health issues or in emergency situations he needs O+ blood group, for his requirements, he will try to check the nearest hospitals with blood availability.

Using the feature called Blood Available in our project he will see the list of hospitals near to him and, also its shows the blood availability, their address, contact information. Below, you can see the list of hospitals near to his location.

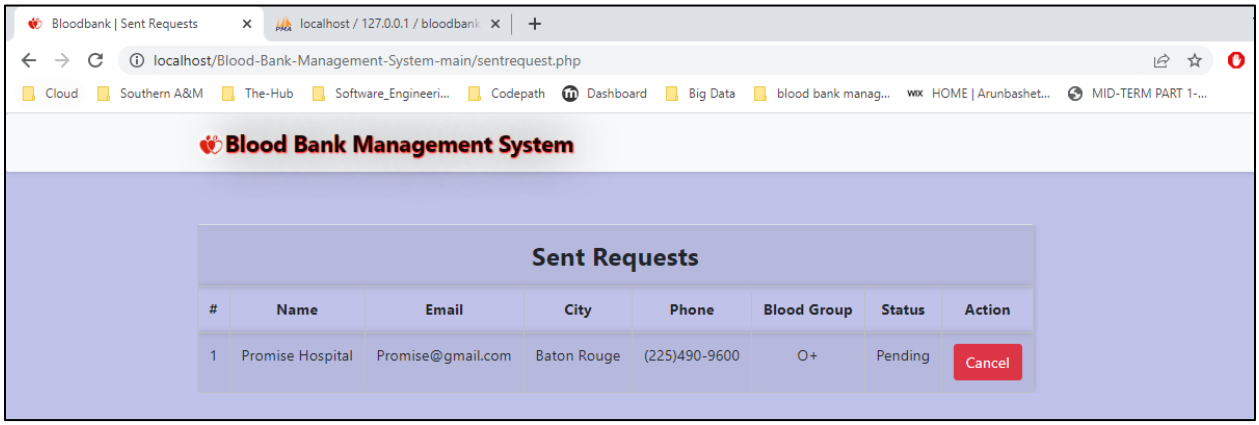


As we knew that he wants O+ blood and for that he can use the above search option and click on search to get the only the hospitals those have O+ blood available.

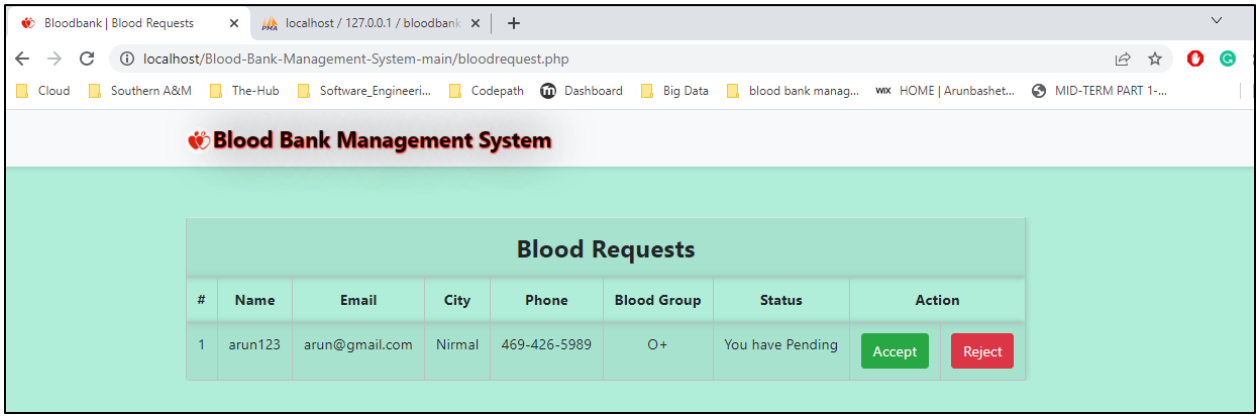


Luckily, Promise Hospital has O+ blood available and user will Request Sample by clicking on Request Sample button. User will see You have requested for blood group O+. For the updation of your request you can check your Status now.

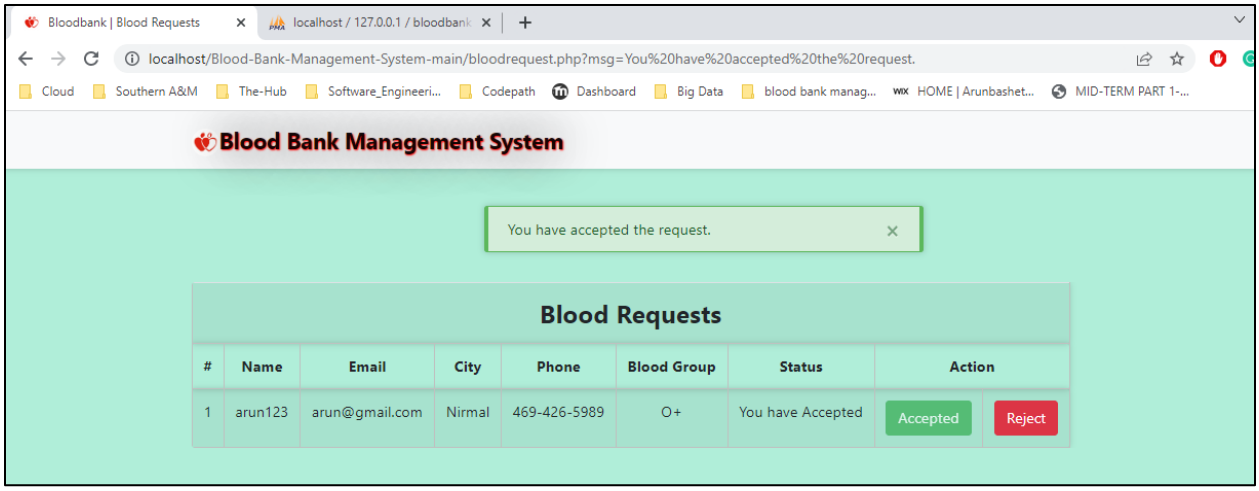
Now using the featured called status of request, they can be able to check weather they accepted the request or rejected. Now, the status is Pending because the Promise Hospital is not yet accepted or rejected.



I will login into Promise Hospital account and check whether they have received the O+ blood notification or not.

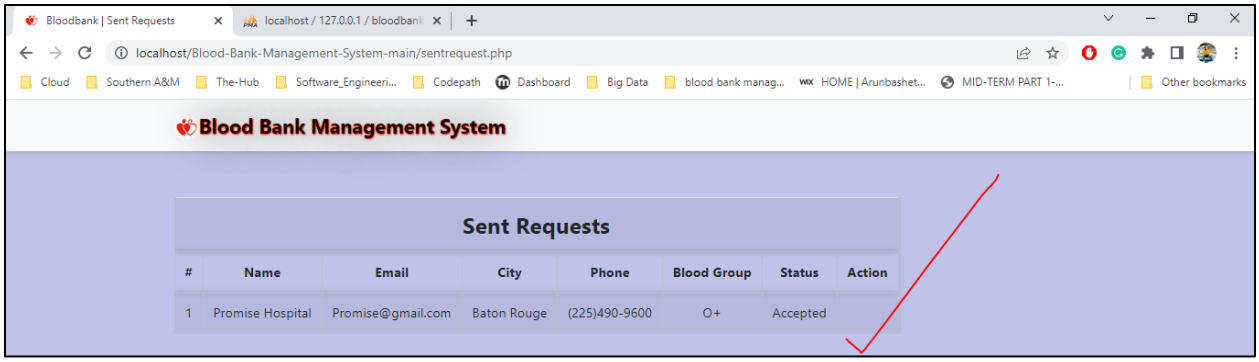


If you see above, user Arun requested for O+ and Promise hospital can either accept it or reject it, I will accept now, you can be able to see the notification “you have accepted the request”



Now, again I will go to into the Arun User account and, check whether they got the notification from Promise hospital that they have accepted to provide the blood.

If you see below screenshot is showing that Donor/Receiver able to see the accepted notification on their account.



Source code is Uploaded to my GIT repository and here is the link to view it.

https://github.com/Bashetty515/Blood-Bank-Management-System-SUBR-Project

My UNIQUE Distribution for this Project:

I am Team leader for this Project, I involved in all the Phases such as Requirements, Specifications, Design, Implementation. I have discussed with my teammates for testing stage which contain Verification and Validation. Apart from the Software model phases, always attended the team meeting to provide the best solutions and, also accepted the innovators innovative ideas.

Particularly, for the Requirement Phase, I involved while discussing about the software and Hardware requirements and, Delimitations.

Particularly, for the Specification Phase, I involved in while discussing about the UML Diagram and Rapid Prototype.

For the Design Phase, I have discussed with the team members for which type of software we will going to use for drawing the **class and sequence diagram.**

For the Implementation part, I have fixed some code and provide the connection between frontend and backend with PHP.